Σχεδιασμός Εφαρμογών Διαδικτύου με Εργαλεία Ανοικτού Λογισμικού

PHP – MySQL

Εγχειρίδιο

Κατσάλης Νίκος
Καθηγητής ΠΕ 19
ΜΔΕ στην Επιστήμη Υπολογιστών

Περιεχόμενα

1 Εισαγωγή 4
1.1 Σκοπός Εγχειριδίου 4
1.2 Προαπαιτούμενες γνώσεις 4
2 HTML και Προγραμματισμός 5
2.1 Μοντέλο επικοινωνίας στον Παγκόσμιο ιστό 5
2.2 Η γλώσσα HTML 5
2.3 Προγραμματισμός από την πλευρά του πελάτη 6
2.4 Προγραμματισμός από την πλευρά του Εξυπηρετητή 7
3 Τι χρειάζομαι για την PHP-MySQL 8
4 MySQL 9
4.1 Βασικές Ενέργειες Χρήσης 9
4.2 Περιβάλλοντα Επικοινωνίας 9
4.3 MySQL και PHP 9
5 Η γλώσσα PHP 10
5.1 Τι είναι η PHP 10
5.2 Πως λειτουργεί η PHP 11
5.2.1 Βασικός Τρόπος Λειτουργίας 11
5.2.2 Δομή συνολικού συστήματος 11
5.3 Κανόνες σύνταξης – Εντολές 12
5.3.1 Βγαίνοντας από την HTML και μπαίνοντας στον κώδικα PHP 12
5.3.2 Διαχωρισμός εντολών και σχόλια 13
5.4 Τύποι Δεδομένων 13
5.5 Μεταβλητές και Σταθερές 15
5.5.1 Μεταβλητές 15
5.5.2 Σταθερές 18
5.6 Τελεστές και Εκφράσεις 18
5.6.1 Τελεστές 18
5.6.2 Εκφράσεις 21
5.7 Δομές Ελέγχου 21
5.7.1 If – If-else - If-elseif-else 21
5.7.2 switch 22
5.7.3 while 22
5.7.4 do..while 23
5.7.5 for 23
5.7.6 foreach 23
5.7.7 break και continue 23
5.8 require(), require_once() και include_once(), include() 24
5.9 Συναρτήσεις 24
5.9.1 Συναρτήσεις οριζόμενες από τον χρήστη 24
5.9.2 Ενσωματωμένες Συναρτήσεις 26
5.10 Κλάσεις και Αντικείμενα 26
5.11 Παραπομπή Συναρτήσεων 27
5.11.1 Συχνά Χρησιμοποιούμενες Ενσωματωμένες Συναρτήσεις 28
5.11.2 Συναρτήσεις του MySQL Module 29
5.12 Υπερκαθολικές (Superglobal) μεταβλητές 31
5.12.1 Αποστολή δεδομένων από τον πελάτη/χρήστη 32
5.12.2 Πληροφορίες Εξυπηρετητή, Πελάτη και Περιβάλλοντος 33
5.13 Μεθοδολογία Σύνδεσης PHP-MySQL 33
5.13.1 Σύνδεση με τον MySQL εξυπηρετητή 33

5.13.2 Σύνδεση με τη Βάση Δεδομένων του εξυπηρετητή 33
5.13.3 Εκτέλεση Ερωτημάτων 33
5.13.4 Έλεγχος και εκτύπωση αποτελεσμάτων ερωτημάτων 34
5.13.5 Κλείσιμο της σύνδεσης 34
5.13.6 Ολοκληρωμένο παράδειγμα των βημάτων 34
5.14 Παραδείγματα Εφαρμογών 36
5.14.1 Παράδειγμα 1 36
5.14.2 Παράδειγμα 2 37
5.14.3 Παράδειγμα 3 40
6 Αναφορές 43

1 Εισαγωγή

Εδώ και πολλά χρόνια, η χρήση του παγκόσμιου ιστού έχει περάσει από την παρουσίαση στατικού περιεχομένου, μέσω στατικών ιστοσελίδων σε γλώσσα HTML, στην παρουσίαση δυναμικού περιεχομένου με βάση τις ενέργειες που κάνει ο χρήστης, τα δεδομένα που αποστέλλει, στοιχεία που υπάρχουν σε Βάσεις Δεδομένων και διάφορες εφαρμογές (e-mail servers κλπ) αλλά και με το ποιος είναι ο ίδιος ο χρήστης.

Με τον τρόπο αυτό δημιουργήθηκε η δυνατότητα για την παροχή νέων υπηρεσιών από τον παγκόσμιο ιστό, οι οποίες συνδέονται με real-time δεδομένα τα οποία βρίσκονται αποθηκευμένα (συνήθως σε Β.Δ.) και μπορεί ο χρήστης να διαχειριστεί. Ενδεικτικές τέτοιες υπηρεσίες περιλαμβάνουν το e-banking, e-commerce, e-government, blogs κλπ, και έχουν δώσει μία νέα διάσταση διαδραστικότητας και ζωντανού περιεχομένου στο Διαδίκτυο.

Η δυνατότητα για τη παρουσίαση δυναμικού περιεχομένου είχε ξεκινήσει αρκετά χρόνια πριν από την πλευρά του εξυπηρετητή με τη χρήση **CGI** (**Common Gateway Interface**), αλλά τα χρόνια που έχουν ακολουθήσει είχαμε μια δραματική εξέλιξη σε νέες τεχνολογίες και γλώσσες που επιτρέπουν τη δημιουργία δυναμικού περιεχομένου στην πλευρά του εξυπηρετητή όπως **JSP**, **ASP**, **PHP** κλπ. Επίσης με την χρήση γλωσσών προγραμματισμού που εκτελούνται στην πλευρά του πελάτη (**VBScript**, **Javascript** κλπ) έχουμε τη δυνατότητα για σημαντικά καλύτερα αποτελέσματα, στον έλεγχο και τη διαδραστικότητα της επικοινωνίας.

Σήμερα υπάρχουν έτοιμες λύσεις που χρησιμοποιούν τις παραπάνω τεχνολογίες και παρέχουν ολοκληρωμένα περιβάλλοντα διαχείρισης στατικού και δυναμικού περιεχομένου, χρηστών, επιπέδων ασφαλείας κλπ.

Το σημαντικό είναι ότι πολλές από αυτές τις τεχνολογίες, γλώσσες αλλά και ολοκληρωμένα περιβάλλοντα είναι λογισμικό **Ανοικτού-Κώδικα** (**open-source**), παρέχοντας νέες διαστάσεις στην εξέλιξή τους και τη χρήση τους στο διαδίκτυο.

1.1 Σκοπός Εγχειριδίου

Σκοπός του εγχειριδίου αυτού είναι να σας παρουσιάσει την χρήση της γλώσσας **PHP** (**Hypertext Preprocessor**) για την παραγωγή δυναμικών σελίδων στην πλευρά του εξυπηρετητή πριν την αποστολή της στον πελάτη που έκανε την αίτηση.

Το παρόν εγχειρίδιο δεν αποτελεί πλήρη οδηγό χρήσης της PHP αλλά επικεντρώνεται κυρίως στην χρήση της PHP σε συνεργασία με Βάσεις Δεδομένων MySQL. Γι' αυτό παρουσιάζονται τα βασικά στοιχεία της γλώσσας PHP (συντακτικό, μεταβλητές, δομές κλπ) και ένα μικρό μέρος από τις έτοιμες συναρτήσεις που υπάρχουν στις βιβλιοθήκες της και οι οποίες αφορούν την σύνδεσή της με βάσεις Δεδομένων MySQL. Μετά ακολουθεί ένας οδηγός για το πώς γίνεται η σύνδεση με MySQL μέσα από τον κώδικα PHP και παρουσιάζονται παραδείγματα εφαρμογών στις οποίες εκτελούνται ερωτήματα σε ΒΔ ενός MySQL εξυπηρετητή στηριζόμενα σε δεδομένα που παρέχει ο χρήστης-πελάτης και δημιουργούνται δυναμικές σελίδες από τα αποτελέσματα των ερωτημάτων, οι οποίες του αποστέλλονται.

1.2 Προαπαιτούμενες γνώσεις

Το εγχειρίδιο θεωρεί ότι ο αναγνώστης έχει βασικές γνώσεις όσον αφορά τον προγραμματισμό (αλγορίθμους, δομές δεδομένων κλπ), της γλώσσας ερωταποκρίσεων SQL καθώς και αρκετά καλή γνώση της γλώσσας HTML. Επίσης χρήσιμο είναι και η γνώση χρήσης κάποιου περιβάλλοντος επικοινωνίας με τον MySQL για τη δημιουργία Β.Δ., πινάκων και χρηστών.

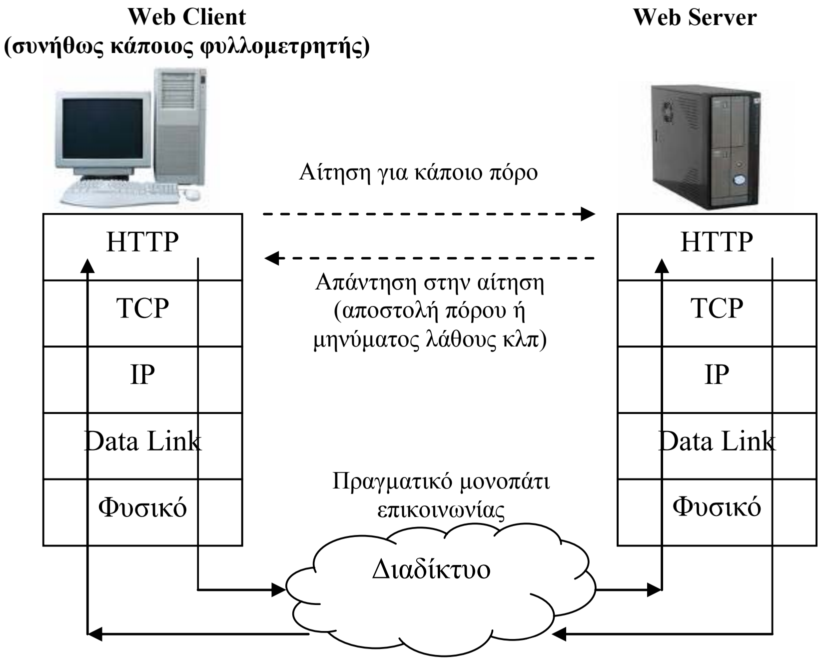
2 HTML και Προγραμματισμός

2.1 Μοντέλο επικοινωνίας στον Παγκόσμιο ιστό

Το μοντέλο επικοινωνίας στον Παγκόσμιο ιστό ακολουθεί το μοντέλο Πελάτη-Εξυπηρετητή (Client-Server Model). Το πρωτόκολλο που χρησιμοποιείται για την επικοινωνία είναι το **HTTP (HyperText Transfer Protocol)** και παρέχει τη δυνατότητα στο πελάτη να κάνει μία αίτηση για κάποιον **πόρο** (**resource**) και στον εξυπηρετητή να απαντήσει στον πελάτη με κάποιο μήνυμα ή στέλνοντας του τον πόρο που ζήτησε. Ο τρόπος που καθορίζει (η μορφή της σύνταξης) ο πελάτης τον πόρο είναι με την χρήση του **URL** (**Uniform Resource Locator** - **Ομοιόμορφος Εντοπισμός Πόρου**). Όταν λέμε **πόρος** (**resource**) αναφερόμαστε σε οποιοδήποτε είδος πληροφορίας (συνήθως ως ένα αρχείο), όπως μία ιστοσελίδα, μία εικόνα, ένα video, ένα αρχείο ήχου κλπ. Αυτός ο πόρος συνήθως βρίσκεται σε κάποιο μέσο αποθήκευσης του Η/Υ στον οποίο τρέχει εξυπηρετητή, αλλά μπορεί να βρίσκεται και οπουδήποτε αλλού.

Πελάτης είναι συνήθως ένα πρόγραμμα περιήγησης ιστοσελίδων (**Φυλλομετρητής**-**Web Browser**) σε κάποιον Η/Υ. Παραδείγματα πελατών είναι ο **MSIE**, ο **Navigator**, ο **Mozilla Firefox**, ο **Opera** κλπ. Εξυπηρετητής ένα πρόγραμμα εξυπηρέτησης ιστοσελίδων (**Web Server**) σε κάποιον (συνήθως άλλο) Η/Υ. Παραδείγματα εξυπηρετητών είναι ο **MS IIS**, ο **Apache**, ο **Netscape Enterprise Server** κλπ.

Στο επόμενο σχήμα φαίνεται το μοντέλο επικοινωνίας (με προβολή των πρωτοκόλλων που μεσολαβούν, στην περίπτωση χρήσης HTTPS μεσολαβεί, πριν το TCP, το SSL ή TLS).



Μοντέλο επικοινωνίας Παγκόσμιου Ιστού

2.2 Η γλώσσα HTML

Η γλώσσα HTML είναι μια γλώσσα σήμανσης με την οποία περιγράφουμε τη δομή και την μορφή ενός εγγράφου. Η βασική ιδέα δημιουργίας της είναι ότι τα περισσότερα έγγραφα έχουν κοινά στοιχεία (επικεφαλίδες, παραγράφους, πίνακες, λίστες κλπ). Ένα αρχείο HTML είναι ένα απλό αρχείο κειμένου το οποίο μπορούμε να γράψουμε με οποιονδήποτε απλό συντάκτη κειμένου (αν

και υπάρχουν πολύ καλοί ειδικοί συντάκτες για HTML). Τα αρχεία αυτά κειμένου αποτελούνται από:

1. Το κείμενο του εγγράφου
2. Ετικέτες (tags) της HTML

Τα αρχικά HTML προέρχονται από τις λέξεις **Hypertext Markup Language** (**Γλώσσα Σήμανσης Υπερ-κειμένου**).

Δυστυχώς με την HTML δεν μπορούμε να προγραμματίσουμε, δηλαδή το έγγραφο πρέπει να δημιουργηθεί σε HTML και να το έχει ο Web Εξυπηρετητής για να το στέλνει σε όποιον πελάτη του το ζητάει. Από τη στιγμή που δημιουργηθεί το έγγραφο παραμένει ως έχει μέχρι να το αλλάξουμε εμείς. Αυτό έχει ως αποτέλεσμα **να μπορούμε να παρέχουμε μόνο στατικό περιεχόμενο** στους πελάτες, κάτι που είναι σημαντικό αλλά περιορίζει πολύ τις δυνατότητες του παγκόσμιου ιστού (όσον αφορά τις εφαρμογές στις οποίες μπορεί να χρησιμοποιηθεί). Γι' αυτό από νωρίς υπήρχε η απαίτηση να παρέχεται **δυναμικό περιεχόμενο** στον πελάτη και η απαίτηση αυτή οδήγησε στον **προγραμματισμό από την πλευρά του πελάτη** και στον **προγραμματισμό από την πλευρά του εξυπηρετητή** (με τον οποίο ασχολούμαστε στο συγκεκριμένο εγχειρίδιο).

Άλλο ένα πρόβλημα της HTML είναι ότι παρέχει ένα σύνολο σταθερών προκαθορισμένων ετικετών (παρά τις πολλές επεκτάσεις που έχουν γίνει) με τις οποίες δεν μπορείς να περιγράψεις τη δομή και το περιεχόμενο πολύπλοκων εγγράφων ή άλλων δεδομένων (πχ δεν μπορείς να περιγράψεις μια μουσική παρτιτούρα, τη δομή και τα περιεχόμενα μίας αποθήκης, ενός πίνακα μιας ΒΔ, μία μαθηματική εξίσωση κλπ). Έτσι δημιουργήθηκε η **XML** (**eXtensible Markup Language – Επεκτάσιμη Γλώσσα Σήμανσης**) η οποία παρέχει εξελιγμένες δυνατότητες δόμησης και περιγραφής περιεχομένου.

2.3 Προγραμματισμός από την πλευρά του πελάτη

Στον προγραμματισμό από την πλευρά του πελάτη, η ιστοσελίδα HTML εμπλουτίζεται με κατάλληλο κώδικα γραμμένο σε κάποια γλώσσα προγραμματισμού και ο οποίος εκτελείται στον υπολογιστή του πελάτη από το πρόγραμμα περιήγησης ιστοσελίδων (Φυλλομετρητής-Web Browser) που χρησιμοποιεί. Ο κώδικας αυτός έχει συνήθως στόχο:

1. Την παροχή δυναμικών εφέ στην ιστοσελίδα.
2. Τον χειρισμό συμβάντων
3. Την αύξηση και τον έλεγχο της διαδραστικότητας μεταξύ του χρήστη και της ιστοσελίδας

Οι γλώσσες προγραμματισμού που χρησιμοποιούνται στον προγραμματισμό από την πλευρά του πελάτη είναι γλώσσες συγγραφής σεναρίων (scripting languages) και με τη βοήθεια εντολών, συναρτήσεων, αντικειμένων και συμβάντων στα αντικείμενα αυτά, παρέχουν:

1. Τη δυνατότητα για τον χειρισμό των αντικειμένων του προγράμματος περιήγησης ιστοσελίδων.
2. Τον χειρισμό των αντικειμένων της ιστοσελίδας (φόρμες, κουμπιά, πίνακες κλπ)
3. Την επεξεργασία των δεδομένων που εμφανίζει η ιστοσελίδα και τη διεξαγωγή υπολογισμών και μετασχηματισμών σε αυτά τα δεδομένα.

Οι πιο γνωστές από αυτές τις γλώσσες είναι η **Javascript** και η **VBScript**. Αυτές οι γλώσσες μαζί με τη χρήση διαδοχικών φύλλων στυλ (**CSS - Cascading Style Sheets**) και HTML μας δίνουν την λεγόμενη **DHTML** (**Dynamic HTML**).

2.4 Προγραμματισμός από την πλευρά του Εξυπηρετητή

Ο προγραμματισμός από την πλευρά του πελάτη παρουσιάζει μία σειρά περιορισμούς που δεν επιτρέπουν την χρήση του σε εφαρμογές συγκεκριμένων τομέων που απαιτούν πρόσβαση σε συγκεκριμένους πόρους (όπως Βάσεις Δεδομένων και αρχεία που βρίσκονται στον Web εξυπηρετητή ή σε κάποιον άλλο εξυπηρετητή, εφαρμογές που τρέχουν σε συγκεκριμένους εξυπηρετητές με περιορισμένη πρόσβαση κλπ). Επίσης επειδή ο κώδικας εκτελείται στον υπολογιστή του πελάτη (από το συγκεκριμένο πρόγραμμα περιήγησης που διαθέτει), περιορίζεται και από ένα σύνολο άλλων παραγόντων όπως το είδος (εταιρεία αλλά και έκδοση) του προγράμματος περιήγησης και των προτιμήσεων που έχει θέσει σε αυτό ο χρήστης (ασφάλεια, λήψη περιεχομένου κλπ). Για παράδειγμα η VBScript δεν υποστηρίζεται από όλα τα προγράμματα περιήγησης και η Javascript μπορεί να έχει απενεργοποιηθεί στις ρυθμίσεις.

Με όλους αυτούς τους περιορισμούς εφαρμογές τομέων όπως το e-commerce, e-banking, e-government, καθώς και γενικότερης πρόσβασης σε περιορισμένες Βάσεις Δεδομένων (πχ Ιατρικές, Νομικές Β.Δ., Ηλεκτρονικές Βιβλιοθήκες, Β.Δ. Ειδήσεων κλπ) δεν μπορούν να υλοποιηθούν με τον προγραμματισμό από την πλευρά του πελάτη.

Λύση σε όλα αυτά δίνει ο προγραμματισμός από την πλευρά του πελάτη στον οποίο συμβαίνουν τα εξής βήματα:

1. Ο πελάτης ζητάει κάποιον πόρο (με βάση το URL αλλά και άλλες ίσως πληροφορίες που υπάρχουν στο μήνυμα αίτησης του πελάτη) από τον εξυπηρετητή.
2. Ο εξυπηρετητής με βάση τον πόρο που ζητήθηκε εκτελεί κάποιο πρόγραμμα.
3. Το πρόγραμμα αυτό που εκτελεί ο εξυπηρετητής μπορεί να χρησιμοποιήσει ένα σύνολο άλλων πόρων (που βρίσκονται είτε στον Η/Υ του εξυπηρετητή είτε σε άλλους Η/Υ που τρέχουν άλλους εξυπηρετητές), όπως Βάσεις Δεδομένων, Συστήματα Αρχείων, επικοινωνία με άλλα προγράμματα (e-mail, ftp εξυπηρετητές κλπ).
4. Το πρόγραμμα αυτό με την εκτέλεσή του δημιουργεί ως αποτέλεσμα έναν πόρο (συνήθως μία ιστοσελίδα HTML, αλλά μπορεί να είναι οτιδήποτε άλλος συνηθισμένος πόρος) τον οποίο στέλνει ως απάντηση ο εξυπηρετητής.

Οι πιο γνωστές γλώσσες/τεχνολογίες που παρέχουν τη δυνατότητα για προγραμματισμό από την πλευρά του εξυπηρετητή είναι η **PHP**, **ASP**, **JSP** και φυσικά η χρήση **CGI** (με οποιοδήποτε εκτελέσιμο γραμμένο σε οποιαδήποτε γλώσσα προγραμματισμού πχ C, Perl κλπ)

3 Τι χρειάζομαι για την PHP-MySQL

Για να μπορέσετε να χρησιμοποιήσετε PHP με την οποία να έχετε πρόσβαση σε MySQL Βάσεις Δεδομένων, σύμφωνα με το μοντέλο του προγραμματισμού από την πλευρά του εξυπηρετητή χρειάζονται:

1. Ένας εξυπηρετητής Παγκόσμιου Ιστού (Web Server) ο οποίος να υποστηρίζει PHP.
2. Εγκατάσταση της PHP στον συγκεκριμένο εξυπηρετητή με κάποιον τρόπο (πχ ως module ή ως CGI εκτελέσιμο).
3. Εγκατάσταση του MySQL εξυπηρετητή (στο ίδιο ή διαφορετικό υπολογιστή).

Ευτυχώς για τη διευκόλυνσή μας υπάρχουν ολοκληρωμένα πακέτα που εγκαθιστούν όλα τα παραπάνω, και κάνουν τις απαραίτητες ρυθμίσεις, για τα περισσότερα λειτουργικά συστήματα. Στο εγχειρίδιο αυτό προτείνουμε 2 ολοκληρωμένα πακέτα, τόσο για την ευκολία εγκατάστασης και χρήσης αλλά και γιατί ότι εγκαθίσταται είναι λογισμικό Ανοιχτού-Κώδικα (Open-Source).

Το πρώτο είναι το **XAMPP** το οποίο είναι ένα πακέτο που δημιουργήθηκε από την μη κερδοσκοπική ομάδα εργασίας **Apache Friends**, της οποίας στόχος είναι η προώθηση της χρήσης του Apache Web Server. Το πακέτο αυτό εγκαθιστά στον υπολογιστή μας:

1. Τον Apache Web Server (ένας από τους δημοφιλέστερους και καλύτερους web server) με υποστήριξη PHP (με τις απαραίτητες ρυθμίσεις)
2. Τον MySQL Server
3. Τον Filezilla FTP Server

Επίσης το πακέτο περιλαμβάνει και ένα σωρό βοηθητικές εφαρμογές και εργαλεία για τη διαχείριση των εξυπηρετητών (πχ MySQLAdmin, PHPMyAdmin, Webalizer κλπ), καθώς και πολλές έτοιμες εφαρμογές επίδειξης της PHP (Demo PHP Applications). Το πακέτο παρέχεται για αρκετά Λ.Σ. (Linux, Windows, Solaris, MAC OS κλπ) από την ιστοσελίδα http://www.apachefriends.org/en/index.html

Το **Wamp** (**WampServer**) είναι άλλο πολύ καλό πακέτο που μπορεί να χρησιμοποιηθεί και το οποίο περιλαμβάνει:

1. Τον Apache Web Server (ένας από τους δημοφιλέστερους και καλύτερους web server) με υποστήριξη PHP (με τις απαραίτητες ρυθμίσεις)
2. Τον MySQL Server

Και αυτό το πακέτο περιλαμβάνει βοηθητικές εφαρμογές και εργαλεία για τη διαχείριση των εξυπηρετητών (πχ SQLiteManager, PHPMyAdmin). Το πακέτο παρέχεται για το Λ.Σ. των Windows από την ιστοσελίδα http://www.wampserver.com/en/

Επίσης καλό θα ήταν αντί να γράφετε PHP σε έναν απλό κειμενογράφο να χρησιμοποιήσετε έναν ειδικό για PHP κειμενογράφο ο οποίος σας βοηθάει να αποφύγετε πολλά λάθη κατά τη συγγραφή του κώδικα. Τέτοιοι κειμενογράφοι υπάρχουν διαθέσιμοι δωρεάν από διάφορες εταιρείες/οργανισμούς/ομάδες χρηστών.

4 MySQL

Ο MySQL εξυπηρετητής είναι ένας πολύ γρήγορος, πολλών νημάτων (multi-threaded) , πολλών χρηστών (multi-user), σταθερός και ασφαλής εξυπηρετητής Διαχείρισης Βάσεων Δεδομένων SQL (Data Base Management System). Είναι σχεδιασμένος για χρήση σε μεγάλα συστήματα που εκτελούν κρίσιμες και πολύ βαριές λειτουργίες σε πολύ μεγάλες Βάσεις Δεδομένων. Επίσης είναι σχεδιασμένος για να μπορεί να χρησιμοποιηθεί με εύκολο τρόπο είτε ενσωματώνοντας τον σε γενικού σκοπού λογισμικό, είτε επικοινωνώντας πολύ εύκολα με αυτόν, επιτρέποντας έτσι την χρήση του σε ένα τεράστιο εύρος εφαρμογών διαφόρων μεγεθών.

Ο MySQL εξυπηρετητής διατίθεται με διπλή άδεια χρήσης:

1. Μπορεί να χρησιμοποιηθεί ως λογισμικό Ανοικτού-Κώδικα με τους όρους της GNU General Public License (http://www.fsf.org/licenses/).
2. Μπορεί επίσης να αγοραστούν εμπορικές άδειες χρήσης (http://www.mysql.com/company/legal/licensing/).

Η εγκατάσταση του MySQL εξυπηρετητή μπορεί να γίνει με πολύ εύκολο τρόπο σε διάφορα Λ.Σ..

4.1 Βασικές Ενέργειες Χρήσης

Οι Βασικές ενέργειες που κάνει ο χρήστης με τον MySQL είναι η εκτέλεση ενός συνόλου εντολών/queries (ερωτημάτων) με τη γλώσσα SQL (καθώς και άλλες εντολές) που έχουν ως στόχο:

1. Τη δημιουργία και Διαγραφή Βάσεων Δεδομένων.
2. Τη δημιουργία, Διαγραφή και Αλλαγή πινάκων σε μία Β.Δ..
3. Την Επιλογή, Ενημέρωση και Διαγραφή εγγραφών σε πίνακα/πίνακες μιας Β.Δ..
4. Την δημιουργία Χρηστών και την ανάθεση/Αφαίρεση δικαιωμάτων για συγκεκριμένες Β.Δ. και πίνακες Β.Δ..

4.2 Περιβάλλοντα Επικοινωνίας

Το βασικό περιβάλλον επικοινωνίας με τον MySQL εξυπηρετητή γίνεται μέσω γραμμής εντολών όπου ο χρήστης (εφόσον έχει συνδεθεί ως κάποιος χρήστης του εξυπηρετητή με τα κατάλληλα στοιχεία – όνομα χρήστη και κωδικό) πληκτρολογεί την εντολή και αυτή εκτελείται από τον εξυπηρετητή με το κατάλληλο αποτέλεσμα.

Επίσης παρέχονται ένα σύνολο από εργαλεία, τα οποία σου επιτρέπουν να κάνεις τις περισσότερες ενέργειες, που κάνεις από την γραμμή εντολών, χρησιμοποιώντας γραφικό περιβάλλον. Μερικά από αυτά εγκαθίστανται αυτόματα αν έχει εγκατασταθεί ο MySQL εξυπηρετητής με κάποιο από τα ολοκληρωμένα πακέτα που αναφέρθηκαν στην προηγούμενη ενότητα. Φυσικά μπορεί κάποιος να κατεβάσει αυτόνομα τα εργαλεία αυτά και να τα στήσει. Πολλά από αυτά είναι διαθέσιμα από την ιστοσελίδα του MySQL (http://dev.mysql.com) στο σύνδεσμο GUI Tools, ενώ άλλα μπορείς να τα κατεβάσεις από άλλες ιστοσελίδες (πχ το PHPMyAdmin από το http://www.phpmyadmin.com).

4.3 MySQL και PHP

Σκοπός του εγχειριδίου αυτού δεν είναι η εκμάθηση του MySQL αλλά η παρουσίαση της χρήσης του μέσα από την PHP. Ο τρόπος και παραδείγματα για τη χρήση αυτή παρατίθενται στην επόμενη ενότητα που παρουσιάζεται η PHP. Για τη χρήση του MySQL ο αναγνώστης μπορεί να διαβάσει το Εγχειρίδιο Αναφοράς του (http://dev.mysql.com/doc).

5 Η γλώσσα PHP

Στην ενότητα αυτή παρουσιάζονται συνοπτικά τα βασικά στοιχεία της PHP (τι είναι, τρόπος χρήσης, συντακτικό, μεταβλητές, δομές, συχνά χρησιμοποιούμενες συναρτήσεις) καθώς και οι συναρτήσεις, του MySQL Module, για τη σύνδεση με MySQL. Επίσης, στο τέλος της ενότητας παρατίθενται ένας οδηγός για την προτεινόμενη (από το εγχειρίδιο αυτό) μεθοδολογία επικοινωνίας με τον MySQL καθώς και κάποια παραδείγματα έτοιμων προγραμμάτων.

Για πλήρη οδηγό της γλώσσας, το σύνολο των συναρτήσεων και των modules σύνδεσης με άλλες ΒΔ και εφαρμογές, λεπτομέρειες όσον αφορά τη χρήση καθώς και παραδείγματα μπορείτε να δείτε το Εγχειρίδιο Αναφοράς της PHP (http://www.php.net/docs.php).

5.1 *Τι είναι η PHP*

Η PHP είναι μια γενικού σκοπού scripting γλώσσα προγραμματισμού, η οποία είναι ειδικά κατάλληλη για ανάπτυξη εφαρμογών για το Web και μπορεί να ενσωματωθεί στην HTML. Τα αρχικά στο όνομά της αντιπροσωπεύουν το "PHP: Hypertext Preprocessor". Ένα σημαντικό στοιχείο της PHP είναι ότι είναι Ανοιχτού-Κώδικα (Open-Source) με όλα τα πλεονεκτήματα που αυτό προσφέρει. Επίσης η PHP ενώ είναι πολύ απλή για ένα αρχάριο προγραμματιστή συγχρόνως προσφέρει πολλά προχωρημένα χαρακτηριστικά για έναν επαγγελματία προγραμματιστή.

Ο κώδικας PHP είναι εσώκλειστος σε HTML κείμενο σε ειδικά tags (ετικέτες) αρχής και τέλους (δείτε παρακάτω υποενότητα «Βγαίνοντας από την HTML και μπαίνοντας στον κώδικα PHP») που σας επιτρέπουν να μεταφέρεστε μέσα και έξω από το "PHP mode" (PHP τρόπο λειτουργίας).

Παράδειγμα:

Παράδειγμα κώδικα PHP

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php
    echo "<h1>Παράδειγμα PHP script!</h1>";
    ?>
  </body>
</html>
```

Το αποτέλεσμα της εκτέλεσης

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
  <h1>Παράδειγμα PHP script!</h1>
  </body>
</html>
```

Ο κώδικας της PHP εκτελείται στον Web Εξηπηρετητή. Έτσι αν το παραπάνω πρόγραμμα (script) βρισκόταν στον εξυπηρετητή και το ζητούσε ο πελάτης, τελικά θα λάμβανε ένα αρχείο που θα περιείχε το αποτέλεσμα της εκτέλεσης του κώδικα, δηλαδή HTML κείμενο. Μάλιστα ο πελάτης δεν έχει κανέναν τρόπο να καταλάβει τι κώδικας υπάρχει από πίσω (αν φροντίσουμε το αρχείο να έχει κατάληξη .html και να έχουν γίνει οι κατάλληλες ρυθμίσεις στον εξυπηρετητή ώστε να εκτελεί PHP και σε αρχεία με κατάληξη .html).

Με την χρήση της PHP, λοιπόν, αντί να γράφουμε κάποιο πρόγραμμα σε κάποια γλώσσα προγραμματισμού το οποίο να παράγει σαν έξοδο ένα HTML αρχείο, γράφουμε ένα HTML αρχείο μέσα στο οποίο όποτε θέλουμε γράφουμε PHP κώδικα ο οποίος τυπώνει HTML στο σημείο του HTML αρχείου που γράφουμε τον κώδικα. Φυσικά με την PHP δεν είμαστε περιορισμένοι να εξάγουμε μόνο HTML αλλά και οποιοδήποτε άλλο κείμενο, XHTML, XML, εικόνες, αρχεία PDF, ακόμη και ταινίες Flash (χρησιμοποιώντας τα libswf και Ming). Επίσης όλα αυτά μπορεί να εξάγονται και να αποθηκεύονται στο σύστημα αρχείων. Επίσης ένα από τα μεγαλύτερα πλεονεκτήματα της PHP είναι η δυνατότητά της για επικοινωνία με ένα μεγάλο πλήθος Συστημάτων Διαχείρισης Βάσεων Δεδομένων όπως:

Adabas D	Ingres	Oracle (OCI7 and OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

Παρόλο που η κύρια χρήση της PHP, όπως ήδη αναφέρθηκε, είναι για εκτέλεση προγραμμάτων (scripts) στον εξυπηρετητή, μπορεί να κάνει και πολλά άλλα όπως:

- **Command line scripting** με τη χρήση του μεταγλωττιστή της PHP (χωρίς Web εξυπηρετητή και Web browser)
- **Client-side GUI** με τη χρήση του **PHP-GTK**.

Το συγκεκριμένο εγχειρίδιο περιορίζεται στην χρήση της μόνο για εκτέλεση προγραμμάτων (scripts) στον εξυπηρετητή. Για τις άλλες χρήσης μπορείτε να πάρετε πληροφορίες από την ιστοσελίδα της PHP (http://www.php.net).

5.2 Πως λειτουργεί η PHP

5.2.1 Βασικός Τρόπος Λειτουργίας

Όπως ήδη αναφέραμε η PHP είναι βασικά μια scripting γλώσσα προγραμματισμού που εκτελείται στον εξυπηρετητή (τουλάχιστον αυτό μας ενδιαφέρει στο συγκεκριμένο εγχειρίδιο). Η γενική της λειτουργία είναι η εξής:

1. Ένας πελάτης ζητάει μια ιστοσελίδα από τον εξυπηρετητή
2. Εφόσον η συγκεκριμένη ιστοσελίδα έχει κατάληξη (πχ .php, .htm κλπ) που εμείς έχουμε ορίσει ότι θέλουμε να ελέγχεται από τον PHP parser/interpreter (ο διερμηνευτής, δηλαδή το πρόγραμμα που διατρέχει την ιστοσελίδα και εκτελεί τα scripts PHP που βρίσκει) πριν αποσταλεί στον χρήστη δίνεται στον PHP parser/interpreter.
3. Το αποτέλεσμα του PHP parser/ interpreter αποστέλλεται στον πελάτη.

Ανάλογα με το Λειτουργικό Σύστημα του εξυπηρετητή καθώς και το είδος του εξυπηρετητή υπάρχουν πολλοί τρόποι εγκατάστασης και λειτουργίας της PHP. Οι πιο γνωστοί είναι:

- Σαν ένα module
- Σαν ένα CGI executable

Η εγκατάσταση σαν module είναι η πιο αποδοτική και ασφαλής.

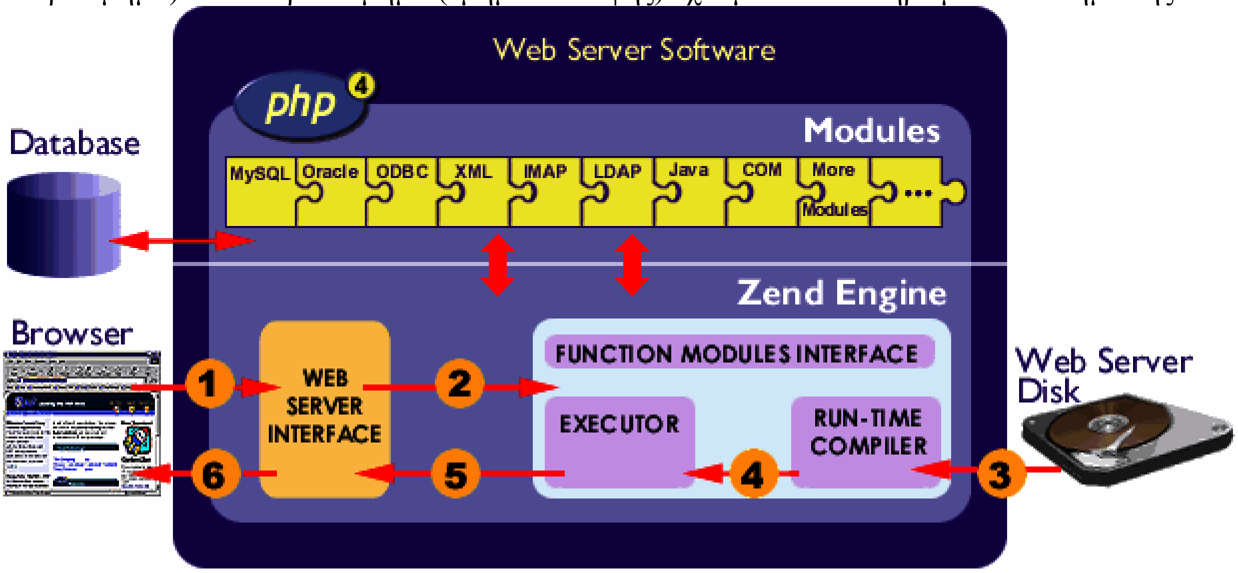
5.2.2 Δομή συνολικού συστήματος

Η δομή της PHP όπως φαίνεται από έξω, αν τη δούμε σαν συνολικό σύστημα, καθώς και τα βήματα λειτουργίας (1, 2, 3 …) παρουσιάζονται στο επόμενο σχήμα.

Το βασικό κομμάτι στο οποίο στηρίζεται η PHP είναι η μηχανή της γλώσσας (language engine) η οποία ονομάζεται Zend και ουσιαστικά αποτελεί τον πυρήνα της. Για να δημιουργηθεί ένας λειτουργικός Web server interpreter (διερμηνευτής) απαιτούνται τρία τμήματα:

1. Το **τμήμα του διερμηνευτή** (***interpreter part)*** το οποίο αναλύει τον κώδικα, τον μεταγλωττίζει και τον εκτελεί.
2. Το **τμήμα λειτουργιών** (***functionality part***) που υλοποιεί η λειτουργικότητα της γλώσσας (συναρτήσεις κλπ).
3. The **τμήμα διεπαφής** (***interface part***) που επικοινωνεί με τον Web server.

Όπως φαίνεται και στο σχήμα η μηχανή Zend περιλαμβάνει όλο το πρώτο τμήμα (διερμηνευτή) και ένα μέρος του δεύτερου (τμήμα λειτουργιών), κυρίως το βασικό τμήμα των λειτουργιών της γλώσσας και των ενσωματωμένων συναρτήσεων. Μαζί με τα modules επέκτασης (το υπόλοιπο δεύτερο τμήμα) και το τρίτο τμήμα (τμήμα διεπαφής) έχουμε το ολοκληρωμένο σύστημα της PHP.



Η δομή του συστήματος της PHP

5.3 Κανόνες σύνταξης – Εντολές

5.3.1 Βγαίνοντας από την HTML και μπαίνοντας στον κώδικα PHP

Υπάρχουν διάφοροι τρόποι για να καθορίσεις πως οριοθετείτε (αρχίζει και τελειώνει) ο κώδικας PHP μέσα σε ένα αρχείο HTML. Ο καλύτερος τρόπος είναι χρησιμοποιώντας τα tags:

```
<?php          . . . εδώ μπαίνει ο κώδικας . . .      ?>
```

Δηλαδή με το tag **<?php** δηλώνουμε ότι ξεκινάει ο κώδικας και με το **?>** ότι τελείωσε. Μόλις ο parser της PHP συναντήσει την αρχή, ξεκινάει την εκτέλεση του κώδικα μέχρι να συναντήσει το κλείσιμο.

Ο τρόπος αυτός είναι ο καλύτερος γιατί είναι πάντα διαθέσιμος και υποστηρίζεται και σε κώδικα συμβατό με XML όπως XHTML.

Άλλοι τρόποι είναι με τα tags

```
<script language="php">   . . . εδώ μπαίνει ο κώδικας . . .      </script>
```

Επίσης μπορεί να χρησιμοποιηθούν ASP-style tags **<%** για άνοιγμα και **%>** για κλείσιμο αλλά πρέπει να ενεργοποιηθούν με κατάλληλες ρυθμίσεις.

Παραδείγματα κώδικα με τη χρήση των δύο βασικών τρόπων:

```
<?php echo("if you want to serve XHTML or XML documents, do like this\n");  ?>
<?php echo ("this is the simplest, an SGML processing instruction\n"); ?>
<script language="php">
echo ("some editors (like FrontPage) don't like processing instructions");
```

```
</script>
<% echo ("You may optionally use ASP-style tags"); %>
```

Φυσικά στην PHP μπορείς να κάνεις και πιο προχωρημένο κώδικα όπως:

```
<?php
if ($expression) {
   ?>
   <strong>This is true.</strong>
   <?php
} else {
   ?>
   <strong>This is false.</strong>
   <?php
}
?>
```

Στο παραπάνω παράδειγμα αν η μεταβλητή $expression είναι αληθής τότε θα μείνει το:

```
<strong>This is true.</strong>
```

ενώ αν είναι ψευδής θα μείνει το:

```
<strong>This is false.</strong>
```

5.3.2 Διαχωρισμός εντολών και σχόλια

Ο διαχωρισμός εντολών στην PHP γίνεται όπως και στην C, με τη χρήση του ερωτηματικού ;

Για να γράψουμε σχόλια υποστηρίζονται ότι και στη C, στη C++ και στο Unix shell. Δηλαδή:

```
/* ….Σχόλιο μίας ή περισσότερων γραμμών … */
// η υπόλοιπη γραμμή είναι σχόλιο
# η υπόλοιπη γραμμή είναι σχόλιο
```

Παράδειγμα:

```
<?php
echo "Hello !"; // σχόλιο στυλ c++
echo "Hello again !";
/* Σχόλιο πολλών γραμμών με
                το στυλ της C */
echo "Hello for the third time"; # Σχόλιο στυλ του Unix shell
?>
```

5.4 Τύποι Δεδομένων

Η PHP υποστηρίζει τους εξής πρωταρχικούς τύπους:

1. **boolean (για λογικούς, TRUE ή FALSE)**
2. **integer (για ακεραίους)**
3. **float και double (για πραγματικούς αριθμούς και είναι το ίδιο)**
4. **string (για αλφαριθμητικά)**
5. **array (για πίνακες)**
6. **object (για αντικείμενα)**
7. **Resource (για αναφορές σε εξωτερικούς πόρους)**

Επίσης υπάρχουν και τρεις σύνθετοι τύποι, οι οποίοι χρησιμοποιούνται κυρίως για ορισμό παραμέτρων:

1. **mixed (για παράμετρο που μπορεί να είναι πολλών τύπων)**
2. **number (για παράμετρο που μπορεί να είναι είτε integer είτε float)**
3. **callback (για πραγματικούς)**

Αυτό που πρέπει να κατανοήσουμε είναι ότι στην PHP δεν χρειάζεται να ορίσουμε εμείς τον τύπο μιας μεταβλητής, αλλά αυτός ορίζεται έμμεσα από την τιμή που δίνουμε στην μεταβλητή κατά τη διάρκεια εκτέλεσης του προγράμματος (κοίτα επόμενη υποενότητα «Μεταβλητές»). Γι' αυτό υπάρχουν μια σειρά συναρτήσεις σχετικές με τον έλεγχο και τη μετατροπή τύπων. Μερικές από αυτές είναι:

- **var_dump()**, η οποία παίρνει σαν όρισμα μία ή περισσότερες εκφράσεις και τυπώνει τον τύπο και την τιμή τους.
- **gettype()**, η οποία παίρνει σαν όρισμα μία μεταβλητή και τυπώνει τον τύπο της τυπώνοντας ένα string. Χρησιμοποιείται κυρίως για debugging.
- **is_integer(), is_string(), is_boolean() κλπ**, οι οποίες επιστρέφουν TRUE ή FALSE αν η μεταβλητή που τους δίνεται ως όρισμα είναι του συγκεκριμένου τύπου.
- **cast()** και **settype()**, με τις οποίες κάνουμε μετατροπή τύπων (typecasting)

Παραδείγματα:

```
<?php
$bool = TRUE;   // a boolean
$str  = "foo";  // a string
$int  = 12;     // an integer

echo gettype($bool); // prints out "boolean"
echo gettype($str);  // prints out "string"

// If this is an integer, increment it by four
if (is_int($int)) {
   $int += 4;
}

// If $bool is a string, print it out
// (does not print out anything)
if (is_string($bool)) {
   echo "String: $bool";
}
?>
```

Στην PHP υπάρχει μια ειδική τιμή NULL (που γράφεται και με οποιονδήποτε συνδυασμό μικρών-κεφαλαίων) και η οποία σημαίνει κενό. Την τιμή NULL την οποία παίρνει μια μεταβλητή στις εξής περιπτώσεις:

- Της έχουμε αναθέσει την τιμή NULL
- Δεν της έχουμε αναθέσει ακόμα τιμή
- Έχει κληθεί η συνάρτηση **unset()** με όρισμα τη μεταβλητή

Χρήσιμη συνάρτηση για να ελέγξεις αν κάποια μεταβλητή είναι NULL είναι η **is_null().**

Ο επόμενος πίνακας δείχνει μια σειρά μεταβλητών και τι επιστρέφουν οι πιο γνωστές συναρτήσεις για αυτές:

Expression	gettype()	empty()	is_null()	isset()	boolean : **if($x)**
$x = "";	string	TRUE	FALSE	TRUE	FALSE

Expression	gettype()	empty()	is_null()	isset()	boolean : if($x)
$x = **NULL**	**NULL**	**TRUE**	**TRUE**	**FALSE**	**FALSE**
var $x;	**NULL**	**TRUE**	**TRUE**	**FALSE**	**FALSE**
$x is undefined	**NULL**	**TRUE**	**TRUE**	**FALSE**	**FALSE**
$x = array();	**array**	**TRUE**	**FALSE**	**TRUE**	**FALSE**
$x = false;	**boolean**	**TRUE**	**FALSE**	**TRUE**	**FALSE**
$x = true;	**boolean**	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = 1;	**integer**	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = 42;	**integer**	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = 0;	**integer**	**TRUE**	**FALSE**	**TRUE**	**FALSE**
$x = -1;	**integer**	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = "1";	**string**	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = "0";	**string**	**TRUE**	**FALSE**	**TRUE**	**FALSE**
$x = "-1";	**string**	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = "php";	**string**	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = "true";	**string**	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = "false";	**string**	**FALSE**	**FALSE**	**TRUE**	**TRUE**

5.5 Μεταβλητές και Σταθερές

5.5.1 Μεταβλητές

Οι μεταβλητές στην PHP αναπαρίστανται από το σύμβολο του δολαρίου ακολουθούμενο από το όνομα της μεταβλητής.

Το όνομα της μεταβλητής είναι case-sensitive. Τα ονόματα των μεταβλητών ακολουθούν τους ίδιους κανόνες όπως και οι labels στην PHP. Ένα έγκυρο όνομα μεταβλητής αρχίζει με ένα γράμμα ή underscore, ακολουθούμενο από οποιονδήποτε αριθμό από γράμματα, αριθμούς, ή underscores.

Παραδείγματα:

```
<?php
$var = "Bob";
$Var = "Joe";
echo "$var, $Var";      // outputs "Bob, Joe"

$4site = 'not yet';    // invalid; starts with a number
$_4site = 'not yet';    // valid; starts with an underscore
$tδyte = 'mansikka';    // valid; 'δ' is (Extended) ASCII 228.
?>
```

Στην PHP όταν θέτουμε μια μεταβλητή ίση με μία άλλη τότε απλώς το περιεχόμενο στης δεύτερης αντιγράφεται στην πρώτη. Πχ.

```
<?php $a = 10; $b = $a; ?>
```

Τώρα και η μεταβλητή $b είναι μεταβλητή τύπου Integer με τιμή 10 αλλά είναι μια διαφορετική μεταβλητή από την $a, δηλαδή αν αλλάξουμε την τιμή της $a δεν θα επηρεαστεί η $b.

Στην PHP υποστηρίζεται και η **ανάθεση με αναφορά** στην οποία η διαφορά είναι ότι βάζουμε μπροστά από τη δεύτερη μεταβλητή το & και τότε απλώς η πρώτη μεταβλητή αντιστοιχίζεται στην

δεύτερη. Δηλαδή αλλάζοντας την μία αλλάζει και η άλλη αφού στην ουσία είναι η ίδια μεταβλητή. Πχ

```
<?php $a = 10; $b = &$a; $b = 5; ?>
```

Τώρα και η μεταβλητή $a και η $b έχουν τιμή 5. Αφού βάλαμε την $b να αντιστοιχιστεί στην $a (**ανάθεση με αναφορά**) και συνεπώς όταν αλλάξαμε την $b άλλαξε και η $a. Η **ανάθεση με αναφορά** εφαρμόζεται μόνο ανάμεσα σε μεταβλητές με όνομα. Αν θέλουμε να πάψει μία από αυτές να δείχνουν στην τιμή αυτή πρέπει να χρησιμοποιηθεί η συνάρτηση **unset().** Έτσι αν θέλαμε η $a να πάψει να έχει αντιστοιχιστεί στην συγκεκριμένη τιμή θα γράφαμε **unset($a)**, κάτι όμως που δεν θα επηρέαζε την $b.

Πρέπει να επισημάνουμε ότι η ανάθεση με αναφορά δεν είναι όπως οι δείκτες σε κάποιες γλώσσες (πχ στη C) και γι' αυτό ενώ κάποιες χρήσεις της είναι όπως των δεικτών (πχ πέρασμα σε συνάρτηση τιμής με αναφορά), δεν έχουν την ίδια συμπερ ιφορά σε όλες τις περιπτώσεις. Στην πραγματικότητα αυτό που γίνεται είναι να δημιουργείται ένα alias στον πίνακα συμβόλων (symbol table) και όχι να έχουμε μια νέα μεταβλητή (όπως στους δείκτες) που και αυτή δείχνει στον ίδιο χώρο μνήμης.

Παρατήρηση: Στην PHP υπάρχουν και μια σειρά από δεσμευμένες μεταβλητές, οι οποίες εξαρτώνται από το περιβάλλον στο οποίο εκτελείται η PHP, αλλά η αναφορά τους είναι πέρα από τους σκοπούς του εγχειριδίου αυτού.

5.5.1.1 Μεταβλητές Πίνακα (Arrays)

Στην PHP ο πίνακας είναι μια ιδιαίτερη δομή που στην πραγματικότητα αποτελεί μια «αντιστοίχηση» που αντιστοιχίζει τιμές σε κλειδιά. Τα κλειδιά είναι οι δείκτες του πίνακα και μπορεί να είναι οποιοσδήποτε βασικός τύπος (πχ ακέραιος, string) και οι τιμές είναι τα περιεχόμενα του πίνακα και μπορεί να είναι οτιδήποτε (ακόμα και άλλος πίνακας) χωρίς να είναι υποχρεωτικά του ίδιου τύπου μεταξύ τους.

Αυτή η ιδιαίτερη δομή σου δίνει μεγάλη ευελιξία προγραμματιστικά αφού χρησιμοποιώντας έναν πίνακα μπορείς να υλοποιήσεις άλλες δομές όπως **Συλλογές** (**Collections**), **Λίστες** (**Lists**), **Ουρές** (**Queues**), **Στοίβες** (**Stacks**), **Λεξικά** (**Dictionaries**) κλπ.

Ο τρόποι για τη δημιουργία ενός πίνακα είναι πολλοί. Ο βασικός είναι με τη γλωσσική δομή **array()**. Το συντακτικό είναι:

array([*key* =>] *value* , ...)

Παράδειγμα

```
<?php
$arr = array("foo" => "bar", 12 => true);

echo $arr["foo"]; // bar
echo $arr[12];    // 1
?>
```

Επίσης μπορείς να δημιουργήσεις πίνακα με τον κλασικό τρόπο των άλλων γλωσσών (χρησιμοποιώντας τα []).

Παράδειγμα

```
<?php
//είναι το ίδιο με το προηγούμενο παράδειγμα
$arr ["foo"] = "bar";
$arr [12] = true;
?>
```

Αν ξεχάσεις κάποιο κλειδί αυτό παίρνει άμεσα την τιμή του μεγαλύτερου ακεραίου κλειδιού που υπάρχει ήδη συν 1 ή την τιμή 0 αν δεν υπάρχει μέχρι εκείνο το σημείο κλειδί που να είναι ακέραιος.

Παράδειγμα

```
<?php
$arr = array("foo" => "bar", 25, 5 => 43, "car", 12 => true);

//Δημιουργούνται τα εξής στοιχεία:
// $arr["foo"] = "bar", $arr[0] = 25, $arr[5] = 43, $arr[6] = "car", $arr[12] = true
?>
```

Για να αλλάξεις ένα στοιχείο απλώς του βάζεις στο συγκεκριμένο δείκτη (κλειδί) μια άλλη τιμή. Για να σβήσεις ένα στοιχείο χρησιμοποιείς τη συνάρτηση **unset()** με όρισμα το στοιχείο. Επίσης υπάρχουν και μία σειρά συναρτήσεις που είναι χρήσιμες για πίνακες (**array_values()**, **arsort()**, **asort()**, **array_keys()**, **array_key_exists()** κλπ).

Χρήσιμη για τη διαχείριση πινάκων είναι η δομή **foreach** (βλέπε ενότητα «Δομές Ελέγχου» παρακάτω).

5.5.1.2 Εμβέλεια Μεταβλητών

Στην PHP η εμβέλεια μιας μεταβλητής αναφέρεται στο περιεχόμενο στο οποίο ορίζεται. Αν μια μεταβλητή χρησιμοποιηθεί μέσα σε ένα αρχείο είναι διαθέσιμη σε όλο το αρχείο αυτό καθώς και σε όλα τα αρχεία που αυτό κάνει include. Μία μεταβλητή που ορίζεται μέσα σε μια συνάρτηση είναι τοπική σε αυτή τη συνάρτηση μόνο. Επίσης για να αναφερθούμε μέσα σε μια συνάρτηση, σε μια μεταβλητή που έχει οριστεί έξω από αυτή, πρέπει να χρησιμοποιήσουμε τη λέξη global. Παραδείγματα:

```
<?php
$a = 10; /* global scope */
include "b.inc";

function Test()
{
   echo $a; /* reference to local scope variable */
}
echo $a;
Test();
?>
```

Εδώ η μεταβλητή $a είναι γνωστή σε όλο το αρχείο καθώς και στο αρχείο”b.inc”, αλλά μέσα στη συνάρτηση για να τη χρησιμοποιήσουμε θα έπρεπε να χρησιμοποιήσουμε τη λέξη global. Έτσι στο συγκεκριμένο παράδειγμα θα τυπωθεί από την προτελευταία εντολή (την echo) η τιμή 10 αλλά από την κλήση της Test() δεν θα τυπωθεί τίποτα γιατί η $a μέσα στη συνάρτηση θεωρείται ότι είναι μια άλλη local μεταβλητή. Για να αναφερθούμε στην έξω $a θα έπρεπε η συνάρτηση να γραφεί:

```
function Test()
{
   global $a;
   echo $a; /* reference to the global $a variable */
}
```

Ένας άλλος τρόπος για να χρησιμοποιήσουμε global μεταβλητές είνα να χρησιμοποιήσουμε τον ειδικά ορισμένο από την PHP πίνακα **$GLOBALS**, ο οποίος περιέχει όλες τις global μεταβλητές και μπορούμε να αναφερθούμε στην κάθε μία χρησιμοποιώντας το όνομά της σαν δείκτη του πίνακα. Έτσι θα μπορούσαμε να γράψουμε:

```
function Test()
{
   echo GLOBALS['a']; /* reference to the global $a variable */
}
```

Μάλιστα ο συγκεκριμένος πίνακας **$GLOBALS** είναι διαθέσιμος παντού, δηλαδή είναι ένας superglobal πίνακας.

5.5.2 Σταθερές

Μια σταθερά είναι ένα όνομα που αντιστοιχεί σε μια απλή τιμή και δεν μπορεί να αλλάξει. Όπως φαίνεται και από το όνομα, αυτή η τιμή δεν μπορεί να αλλάξει κατά την εκτέλεση του script. Για το όνομά τους ακολουθούνται οι κανόνες των μεταβλητών.

Όπως οι superglobals, η εμβέλεια (scope) μιας σταθεράς είναι global. Μπορείτε να έχετε πρόσβαση σ' αυτή από οποιοδήποτε μέρος στο script χωρίς να λαμβάνετε υπόψη την εμβέλεια.

Για να οριστεί μια σταθερά μπορεί να χρησιμοποιηθεί η συνάρτηση **define()**. Για να πάρουμε την τιμή μιας σταθεράς αρκεί να γράψουμε το όνομά της (χωρίς $ μπροστά !!) ή να χρησιμοποιήσουμε τη συνάρτηση **constant().** Πχ.

```
<?php
define("CONSTANT", "Hello world.");
echo CONSTANT; // outputs "Hello world."
echo Constant; // outputs "Constant" and issues a notice.
?>
```

5.6 Τελεστές και Εκφράσεις

5.6.1 Τελεστές

Οι τελεστές μοιάζουν με τους τελεστές της C.

5.6.1.1 Τελεστής Ανάθεσης

Ο βασικός τελεστής ανάθεσης είναι ο "=". Αυτός ο τελεστής δεν σημαίνει "ισούται με" αλλά “ανατίθεται σε". Η τιμή μιας έκφρασης ανάθεσης είναι η τιμή που της ανατίθεται. Δηλαδή, η τιμή του "$a = 3" είναι 3. Αυτό σας επιτρέπει να κάνετε μερικά περίπλοκα πράγματα:

```
$a = ($b = 4) + 5; // $a is equal to 9 now, and $b has been set to 4.
```

Πέρα από το βασικό τελεστή ανάθεσης, υπάρχουν "σύνθετοι τελεστές" για όλους τους δυαδικούς αριθμητικούς και αλφαριθμητικούς τελεστές που σας επιτρέπουν να χρησιμοποιήσετε μια έκφραση και στη συνέχεια να θέσετε την τιμή στο αποτέλεσμα της έκφρασης. Πχ:

```
$a = 3;
$a += 5; // sets $a to 8, as if we had said: $a = $a + 5;
$b = "Hello ";
$b .= "There!"; // sets $b to "Hello There!", just like $b = $b . "There!";
```

Φυσικά όπως ήδη αναφέραμε σε προηγούμενη ενότητα («Μεταβλητές») υπάρχει και η ανάθεση με αναφορά.

5.6.1.2 Αριθμητικοί Τελεστές

Παράδειγμα	Όνομα	Αποτέλεσμα
$a + $b	Πρόσθεση	Αποτέλεσμα του $a και του $b.
$a - $b	Αφαίρεση	Διαφορά του $a και του $b.
$a * $b	Πολλαπλασιασμός	Γινόμενο του $a και του $b.
$a / $b	Διαίρεση	Αποτέλεσμα του $a και του $b.
$a % $b	Modulus	Υπόλοιπο του $a διαιρεμένου από το $b.

5.6.1.3 Δυαδικοί Τελεστές

Παράδειγμα	Όνομα	Αποτέλεσμα
$a & $b	And	Τα bits που είναι 1 τόσο στο $a όσο και στο $b, ενεργοποιούνται.
$a | $b	Or	Τα bits που είναι 1 είτε στο $a είτε στο $b, ενεργοποιούνται.
$a ^ $b	Xor	Τα bits που είναι 1 είτε στο $a είτε στο $b, αλλά όχι και στα δύο, ενεργοποιούνται.
~ $a	Not	Τα bits που δεν είναι ενεργοποιημένα στο $a, ενεργοποιούνται, και αντίστροφα.
$a << $b	Shift left	Μετακίνηση των bits του $a κατά $b βήματα προς τα αριστερά (κάθε βήμα σημαίνει "πολλαπλασιασμός επί δύο")
$a >> $b	Shift right	Μετακίνηση των bits του $a κατά $b βήματα προς τα δεξιά (κάθε βήμα σημαίνει "διαίρεση επί δύο")

5.6.1.4 Συγκριτικοί Τελεστές

Παράδειγμα	Όνομα	Αποτέλεσμα
$a == $b	Ισότητα	TRUE αν το $a είναι ίσο με το $b.
$a === $b	Ομοιότητα	TRUE αν το $a είναι ίσο με το $b, και είναι επιπλέον του ίδιου τύπου. (Στην PHP 4 μόνο)
$a != $b	□νισα	TRUE αν το $a δεν είναι ίσο με το $b.
$a <> $b	Όχι ίσα	TRUE αν το $a δεν είναι ίσο με το $b.
$a !== $b	Ανόμοια	TRUE αν το $a δεν είναι ίσο με το $b, ή αν δεν είναι του ίδιου τύπου. (στην PHP 4 μόνο)
$a < $b	Μικρότερο από	TRUE αν το $a είναι ακριβώς μικρότερο από το $b.
$a > $b	Μεγαλύτερο από	TRUE αν το $a είναι αυστηρώς μεγαλύτερο από το $b.
$a <= $b	Μικρότερο από ή ίσο με	TRUE αν το $a είναι μικρότερο από ή ίσο με το $b.
$a >= $b	Μεγαλύτερο από ή ίσο με	TRUE αν το $a είναι μεγαλύτερο από ή ίσο με το $b.

5.6.1.5 Τελεστής Εκτέλεσης

Η PHP παρέχει τα ανάποδα μονά αυτάκια (bakticks) ` ` τα οποία παίζουν τον ρόλο του τελεστή εκτέλεσης, δηλαδή ότι βάλουμε μέσα θα προσπαθήσει να το εκτελέσει σαν εντολή φλοιού. Πχ.:

```
$output = `ls -al`;
```

```
echo "<pre>$output</pre>";
```

Παρόμοια αποτέλεσμα μπορούμε να έχουμε με τη συνάρτηση **shell_exec()**.

5.6.1.6 Τελεστές Αύξησης-Μείωσης

Παράδειγμα	**Όνομα**	**Αποτέλεσμα**
++$a	Προ-αύξηση	Αυξάνει το $a κατά ένα και επιστρέφει το $a.
$a++	Μετά-αύξηση	Επιστρέφει το $a, και μετά αυξάνει το $a κατά ένα.
--$a	Προ-μείωση	Μειώνει το $a κατά ένα, και μετά επιστρέφει το $a.
$a--	Μετά-μείωση	Επιστρέφει το $a, και μετά μειώνει το $a κατά ένα.

5.6.1.7 Λογικοί Τελεστές

Παράδειγμα	**Όνομα**	**Αποτέλεσμα**
$a and $b	And	**TRUE** αν και το $a και το $b είναι **TRUE**.
$a or $b	Or	**TRUE** αν είτε το $a είτε το $b είναι **TRUE**.
$a xor $b	Xor	**TRUE** αν είτε το $a είτε το $b είναι **TRUE**, αλλά όχι και τα δυο.
! $a	Not	**TRUE** αν το $a δεν είναι **TRUE**.
$a && $b	And	**TRUE** αν και το $a και το $b είναι **TRUE**.
$a || $b	Or	**TRUE** αν είτε το $a είτε το $b είναι **TRUE**.

5.6.1.8 Τελεστές για Strings

Υπάρχουν δύο τελεστές για strings (αλφαριθμητικών). Ο πρώτος είναι ο τελεστής σύνδεσης ('.'), ο οποίος επιστρέφει τη σύνδεση των αριστερών και των δεξιών παραμέτρων. Ο δεύτερος είναι ο τελεστής ανάθεσης σύνδεσης ('.='), ο οποίος προσθέτει την παράμετρο της δεξιά πλευρά στην παράμετρο της αριστερής πλευράς. Πχ.:

```
$a = "Hello ";
$b = $a . "World!"; // now $b contains "Hello World!"

$a = "Hello ";
$a .= "World!";     // now $a contains "Hello World!"
```

5.6.1.9 Τελεστές Πινάκων

Ο μόνος τελεστής πίνακα στην PHP είναι ο + τελεστής. Αυτός προσθέτει τη δεξιά μεριά του πίνακα στην αριστερή μεριά, ενώ τα διπλά κλειδιά ΔΕΝ γίνονται overwrite (δεν γράφονται από πάνω). Πχ.:

```
$a = array("a" => "apple", "b" => "banana");
$b = array("a" =>"pear", "b" => "strawberry", "c" => "cherry");
$c = $a + $b;
```

Το αποτέλεσμα θα είναι ο πίνακας $c να έχει τρία στοιχεία, το a και b του πίνακα $a και το c του πίνακα $b. Διατηρούνται οι τιμές για το a και b από τον πίνακα $a γιατί ο συγκεκριμένος τελεστής δεν κάνει override τα κοινά κλειδιά.

5.6.1.10 Προτεραιότητα Τελεστών

Ο παρακάτω πίνακας έχει τους τελεστές και την προτεραιότητα που έχουν:

<table>
<tr><th>Σχετικότητα</th><th>Τελεστές</th></tr>
<tr><td>αριστερή</td><td>,</td></tr>
<tr><td>αριστερή</td><td>or</td></tr>
<tr><td>αριστερή</td><td>xor</td></tr>
<tr><td>αριστερή</td><td>and</td></tr>
<tr><td>δεξιά</td><td>print</td></tr>
<tr><td>αριστερή</td><td>= += -= *= /= .= %= &= |= ^= <<= >>=</td></tr>
<tr><td>αριστερή</td><td>? :</td></tr>
<tr><td>αριστερή</td><td>||</td></tr>
<tr><td>αριστερή</td><td>&&</td></tr>
<tr><td>αριστερή</td><td>|</td></tr>
<tr><td>αριστερή</td><td>^</td></tr>
<tr><td>αριστερή</td><td>&</td></tr>
<tr><td>Χωρίς σύνδεση</td><td>== != === !==</td></tr>
<tr><td>Χωρίς σύνδεση</td><td>< <= > >=</td></tr>
<tr><td>Αριστερή</td><td><< >></td></tr>
<tr><td>Αριστερή</td><td>+ - .</td></tr>
<tr><td>Αριστερή</td><td>* / %</td></tr>
<tr><td>Δεξιά</td><td>! ~ ++ -- (int) (float) (string) (array) (object) @</td></tr>
<tr><td>Δεξιά</td><td>[</td></tr>
<tr><td>Χωρίς σύνδεση</td><td>new</td></tr>
</table>

5.6.2 Εκφράσεις

Οι εκφράσεις (expressions) είναι το πιο σηματικό κομμάτι της PHP. Στην PHP, σχεδόν όλα όσα γράφετε είναι εκφράσεις. Ο απλούστερος και συγχρόνως ο πιο ακριβής τρόπος για να ορίσουμε μια έκφραση είναι "οτιδήποτε έχει τιμή".

5.7 Δομές Ελέγχου

Οι δομές ελέγχου που υποστηρίζονται από την γλώσσα παρουσιάζονται στις επόμενες υποενότητες.

5.7.1 If – If-else - If-elseif-else

Η if στην PHP είναι ίδια με την C, δηλαδή έχει τους παρακάτω τρόπους σύνταξης, ανάλογα με την χρήση που θέλεις να κάνεις:

If

```
if (expr)
    statement
```

ή

```
if (expr)
{
    statements
}
```

If με else

```
if (expr)
{
    statements
}
else
{
    statements
}
```

If με elseif και else

```
if (expr)
{
    statements
} elseif (expr)
{
    statements
}
else
{
    statements
```

```
}
```

Παραδείγματα

```
<?php
if ($a == 5) {
   print "a is equal  5";
   $a = $a + 1
   print "now a is equal 6";
}
?>
```

```
<?php
if ($a == 5) {
   print "a is equal  5";
} else {
   print "a is not equal 5";
}
?>
```

```
<?php
if ($a > 5) {
   print "a is bigger than 5";
} elseif ($a == 5) {
   print "a is equal to 5";
} else {
   print "a is smaller than 5";
}
?>
```

5.7.2 switch

Η switch είναι και αυτή όπως στη C:

```
switch (expr) {
   case value1:
      statements
      break;
   case value2:
      statements
      break;
   default:
      statements
      break;
}
```

Παράδειγμα:

```
switch ($x) {
   case 0:
      print "It is 0";
      break;
   case 1:
      print "It is 1";
      break;
   case 2:
      print "It is 2";
      break;
}
```

5.7.3 while

Η while στην PHP είναι και αυτή ίδια με την C:

```
while (expr)
    statement
```

ή

```
while (expr)
{
    statements
}
```

Παράδειγμα

```
$i = 1;
while ($i <= 10) {
   print $i;
   $i++;
}
```

5.7.4 do..while

Η do … while στην PHP είναι και αυτή ίδια με την C:

```
do {
    statements
} while (expr);
```

Παράδειγμα

```
$i = 1;
do {
   print $i;
   $i++;
} while ($i <= 10);
```

5.7.5 for

Η for στην PHP είναι και αυτή ίδια με την C:

```
for (expr1; expr2; expr3) {
    statements
}
```

Παράδειγμα

```
for ($i = 1; $i <= 10; $i++) {
   print $i;
}
```

5.7.6 foreach

Στην PHP παρέχεται η δομή foreach όπως και στη γλώσσα Perl, η οποία μας επιτρέπει να προσπελαύνουμε εύκολα πίνακες. Φυσικά δεν μπορεί να χρησιμοποιηθεί με απλές μεταβλητές αλλά μόνο με πίνακες.

```
foreach (array_expression as $value) statement
foreach (array_expression as $key => $value) statement
```

Παράδειγμα

```
$arr = array("one", "two", "three");

//Τυπώνω τις τιμές
foreach ($arr as $value) {
   echo "Value: $value<br>\n";
}
//Αλλάζω τις τιμές
foreach ($arr as $k => $value) {
   $arr[$k] = "number " .  $value;
}
```

5.7.7 break και continue

Η break χρησιμοποιείται για τον τερματισμό της εκτέλεσης της τρέχουσας εντολής for, foreach while, do..while καθώς και της δομής switch. Μάλιστα δέχεται και ένα προαιρετικό αριθμό

(όρισμα) αμέσως μετά το λεκτικό break που καθορίζει πόσες εμφωλευμένες δομές θα τερματιστούν.

Η continue χρησιμοποιείται σε δομές επανάληψης για να παρακαμφτεί το υπόλοιπο της τρέχουσας επανάληψης και να συνεχίσει από την αρχή της επόμενης. Μάλιστα δέχεται και αυτή ένα προαιρετικό αριθμό (όρισμα) αμέσως μετά το λεκτικό continue που καθορίζει πόσα επίπεδα εμφωλευμένων επαναλήψεων θα παρακαμφτούν.

Παραδείγματα

```
$i = 0;
while (++$i) {
   switch ($i) {
   case 5:
      echo "At 5<br>\n";
      break 1;  /* Exit only the switch. */
   case 10:
      echo "At 10; quitting<br>\n";
      break 2;  /* Exit the switch and the while. */
   default:
      break;
   }
}
```

```
$i = 0;
while ($i++ < 5) {
   echo "Outer<br>\n";
   while (1) {
      echo "  Middle<br>\n";
      while (1) {
         echo "  Inner<br>\n";
         continue 3;
      }
      echo "This never gets output.<br>\n";
   }
   echo "Neither does this.<br>\n";
}
```

5.8 require(), require_once() και include_once(), include()

Η PHP είναι μία γλώσσα που ευνοεί τη χρήση συναρτήσεων για να φτιάχνεις επαναχρησιμοποιούμενο κώδικα, να γλιτώνεις επανάληψη του κώδικα και να μπορείς να συντηρείς το πρόγραμμά σου ευκολότερα. Για αυτό είναι προτιμότερο να γράφεις τις συναρτήσεις που θα χρησιμοποιείς σε ένα ή περισσότερα ξεχωριστά αρχεία και όχι στο αρχείο της ιστοσελίδας. Όποτε χρειάζεσαι να χρησιμοποιήσεις κάποιες συναρτήσεις αρκεί να «ενσωματώσεις» το αρχείο με τις συναρτήσεις στο αρχείο της ιστοσελίδας σου (στον κώδικα που έχεις στην ιστοσελίδα). Για να γίνει αυτό χρησιμοποιείς τις συναρτήσεις **require()** και **include()**. Παράδειγμα:

```
<?php
require 'general_funcs.php';
include $afile;
include('funcs.php');
?>
```

Οι συναρτήσεις **require_once()** και **include_once()** είναι παρόμοιες με τη διαφορά ότι αν έχει ήδη ενσωματωθεί το αρχείο τότε δεν θα ξαναενσωματωθεί.

5.9 Συναρτήσεις

Η PHP, όπως ήδη αναφέρθηκε, ευνοεί τη χρήση συναρτήσεων για να μπορεί ο κώδικάς μας να έχει όλα τα πλεονεκτήματα που η χρήση τους προσφέρει. Έτσι παρέχεται η δυνατότητα για να γράψει ο χρήστης τις δικές του συναρτήσεις αλλά παρέχεται και ένα μεγάλο σύνολο έτοιμων (ενσωματωμένων) συναρτήσεων μέσω των οποίων γίνονται οι πιο συνηθισμένες λειτουργίες.

5.9.1 Συναρτήσεις οριζόμενες από τον χρήστη

Η σύνταξη για την ορισμό μιας συνάρτησης από τον χρήστη είναι η εξής:

```
function όνομα (παράμετροι)
{
        statements
}
```

Μέσα σε μια συνάρτηση μπορούν να μπουν ότι εντολές υπάρχουν και σε ένα πρόγραμμα της PHP. Μάλιστα μπορούμε μέχρι και να ορίσουμε συναρτήσεις μέσα σε μια συνάρτηση. Φυσικά οι εσωτερικές, στη συνάρτηση, συναρτήσεις θα υπάρξουν μόνο όταν κληθεί η εξωτερική.

Επειδή στις περισσότερες περιπτώσεις σκοπός της συνάρτησης είναι η επιστροφή κάποιας τιμής, τότε πρέπει μία από της εντολές να είναι η:
return τιμή;

Μόλις εκτελεστεί η συγκεκριμένη εντολή, τερματίζεται η συνάρτηση και επιστρέφεται η συγκεκριμένη τιμή, που ακολουθεί τη λέξη return, στο σημείο που κλήθηκε η συνάρτηση. Η τιμή αυτή είναι οποιοσδήποτε τύπος (ακόμα και πίνακας, αντικείμενο κλπ).

Οι παράμετροι που ακολουθούν το όνομα μιας συνάρτησης χρησιμεύουν για να περάσουμε τιμές κατά την κλήση και είναι μια λίστα μεταβλητών ή/και σταθερών που διαχωρίζονται με κόμματα. Για να περάσουμε μεταβλητού αριθμού παραμέτρους γίνεται χρησιμοποιώντας τις συναρτήσεις **func_num_args()**, **func_get_arg()**, και **func_get_args()**.

Το όνομα μιας συνάρτησης είναι όπως ακριβώς το όνομα μιας μεταβλητής αλλά είναι case-insensitive. Η PHP υποστηρίζει φυσικά και αναδρομή.

Πολλές φορές στις συναρτήσεις χρησιμοποιούμε **static** μεταβλητές για να μπορούμε να διατηρούμε τιμές μεταξύ των κλήσεων.

Παραδείγματα:

```
<?php
//Παράδειγμα με ακέραιο
function print_upto($i)
{
        for ($i = 1; $i <= 10; $i++)
                print $i;
}
//Παράδειγμα με πέρασμα παραμέτρου με αναφορά
function concatVer1(&$str1, $str2)
{
   $string .= 'and something extra.';
}
function concatVer2($str1, $str2)
{
   return $str1 . $str2;
}

$a = 'Hello ';
$b = ‘World !!’;
//Έχουν το ίδιο αποτέλεσμα
concatVer1($a, $b);
$a = concatVer2($a, $b);

//Παράδειγμα με πίνακα
$c = array("one", "two", "three");
function printArray($inarr)
{
   foreach ($inarr as $value) {
      echo "Value: $value<br>\n";
   }
}
```

```
//Παράδειγμα με προκαθορισμένη τιμή
function HelloWithDefault($name = 'anonymous')
{
   return "Hello $name.\n";
}
echo HelloWithDefault ();
echo HelloWithDefault ("Νίκο");

//Παράδειγμα με μεταβλητή συνάρτηση
$x = 'HelloWithDefault';
echo $x(); //θα κληθεί η HelloWithDefault ();

//Παράδειγμα με μεταβλητή συνάρτηση
function funcstat($setval = false)
{
   static $a;

   if ($setval)
     $a = 1;
   else
     $a += 1;
   return $a;
}
$initializeVal = TRUE;
funcstat($initializeVal);
funcstat();
funcstat();
echo funcstat(); //τυπώνει 4
?>
```

Όπως αναφέρθηκε και σε προηγούμενη ενότητα, αν η συνάρτηση πρόκειται να χρησιμοποιηθεί σε περισσότερα από ένα αρχεία-ιστοσελίδες, τότε θα ήταν καλύτερο να ορίζεται σε ξεχωριστό αρχείο που θα ενσωματώνεται στον κώδικα της ιστοσελίδας όποτε χρειάζεται κάποια από τις συναρτήσεις που περιέχει.

5.9.2 Ενσωματωμένες Συναρτήσεις

Όπως αναφέρθηκε υπάρχει ένα μεγάλο σύνολο ενσωματωμένων συναρτήσεων από τις οποίες κάποιες ήδη αναφέρθηκαν στις προηγούμενες παραγράφους. Ορισμένες από τις συχνότερα χρησιμοποιούμενες, μαζί με τις συναρτήσεις που παρέχονται από το MySQL Module, παρατίθενται σε επόμενη ενότητα («Παραπομπή Συναρτήσεων»). Η αναλυτική παρουσίασή όλων των συναρτήσεων (ενσωματωμένων και από τα διάφορα modules) είναι πέρα από τους σκοπούς αυτού του εγχειριδίου.

5.10 Κλάσεις και Αντικείμενα

Η PHP είναι μια object-oriented γλώσσα και υποστηρίζει τη δημιουργία αντικειμένων μέσω του γλωσσικού class (κλάση). Μία κλάση είναι μια συλλογή μεταβλητών (ιδιότητες-χαρακτηριστικά του αντικειμένου) και συναρτήσεων (μέθοδοι χειρισμού του αντικειμένου). Για να δημιουργήσεις ένα νέο αντικείμενο χρησιμοποιείς το **new**.

Παράδειγμα

```
<?php
class Cart
{
   var $items;  // Items in our shopping cart

   // Add $num articles of $artnr to the cart
   function add_item ($artnr, $num)
   {
      $this->items[$artnr] += $num;
   }

   // Take $num articles of $artnr out of the cart
   function remove_item ($artnr, $num)
   {
      if ($this->items[$artnr] > $num) {
         $this->items[$artnr] -= $num;
         return true;
      } else {
         return false;
      }
   }
}

$c = new Cart;
$c->add_item("10", 1);

?>
```

Φυσικά υποστηρίζονται constructors καθώς και κληρονομικότητα με τη χρήση της λέξης **extends**.

Παράδειγμα:

```
<?php
class Named_Cart extends Cart
{
   var $owner;

   function Named_Cart()
   {
       $owner = 'Anonymous';
   }
   function set_owner ($name)
   {
      $this->owner = $name;
   }
}
?>
```

5.11 Παραπομπή Συναρτήσεων

Οι ενσωματωμένες συναρτήσεις που παρέχει η PHP είναι πάρα πολλές. Στην επόμενη υποενότητα παρουσιάζονται επιλεκτικά μερικές που χρησιμοποιούνται πολύ συχνά.

Αμέσως μετά παρουσιάζονται συνοπτικά μόνο αυτές που αφορούν τη σύνδεση PHP με MySQL. Σε επόμενη ενότητα («Μεθοδολογία Σύνδεσης PHP-MySQL») θα παρουσιαστεί η χρήση των πιο σημαντικών από αυτές.

5.11.1 Συχνά Χρησιμοποιούμενες Ενσωματωμένες Συναρτήσεις

Μερικές από τις πιο συχνά χρησιμοποιούμενες συναρτήσεις, οι οποίες δεν έχουν ήδη αναφερθεί στις προηγούμενες ενότητες, παρουσιάζονται στον επόμενο πίνακα.

Συνάρτηση	**Χρήση**
count	Μετράει τα στοιχεία μιας μεταβλητής. Ομοίως και η sizeof.
min, max, sin, cos, tan, round, sqrt, pi, pow, log, exp, rand	Μαθηματικές συναρτήσεις
echo	**void echo (string arg1 [, string argn...])** Εμφανίζει ένα ή περισσότερα strings. Αν χρησιμοποιηθούν μονά εισαγωγικά δεν τυπώνεται η τιμή των μεταβλητών αλλά ότι γράφουμε.
print	**int print (string arg)** Εμφανίζει ένα string επιστρέφει 1. Αν χρησιμοποιηθούν μονά εισαγωγικά δεν τυπώνεται η τιμή των μεταβλητών αλλά ότι γράφουμε.
printf	**void printf (string format [, mixed args])** Εμφανίζει ένα μορφοποιημένο string (όπως στη C)
sprintf	**string sprintf (string format [, mixed args])** Επιστρέφει ένα μορφοποιημένο string (όπως στη C)
fprintf	**int fprintf (resource handle, string format [, mixed args])** Τυπώνει σαν την printf αλλά στο resource
sscanf	**mixed sscanf (string str, string format [, string var1])** Διαβάζει από ένα string με βάση το format
flush	**void flush (void)** Σπρώχνει στην έξοδο ότι υπάρχει στον buffer
number_format	**string number_format (float number [, int decimals])** **string number_format (float number, int decimals, string dec_point, string thousands_sep)** Μορφοποιεί κατάλληλα έναν αριθμό
count_chars	**mixed count_chars (string string [, int mode])** Μετρά το πλήθος των εμφανίσεων κάθε χαρακτήρα και το επιστρέφει με ποικίλους τρόπους (ανάλογα την τιμή του mode)
convert_uudecode **convert_uuencode**	Αποκωδικοποιεί ένα uuencoded string Κωδικοποιεί ένα string σε uuencoded
trim **ltrim** **rtrim**	Βγάζει τα κενά από ένα string Βγάζει τα κενά από τα αριστερά ενός string Βγάζει τα κενά από τα δεξιά ενός string
strchr, strstr, strlen, strcmp, strtok, strtolower, strtoupper	Όπως και στη C
substr	**string substr (string string, int start [, int length])** Επιστρέφει ένα μέρος του string
wordwrap	**string wordwrap (string str [, int width [, string break [, boolean cut]]])** Αναδιπλώνει ένα string μετά από έναν αριθμό χαρακτήρων
empty	Ελέγχει αν μια μεταβλητή είναι empty
system	**string system (string command [, int return_var])** Εκτελεί ένα εξωτερικό πρόγραμμα και επιστρέφει το αποτέλεσμά του.
parse_url	array **parse_url** (string url)

	Αναλύει ένα URL και επιστρέφει τα τμήματά του σε πίνακα
ctype_alnum **ctype_alpha** **ctype_cntrl** **ctype_digit** **ctype_lower** **ctype_upper** **ctype_print** **ctype_space** **ctype_punct** **ctype_xdigit**	Παίρνουν σαν όρισμα ένα string και ελέγχουν (επιστρέφουν true ή false) αν περιλαμβάνει μόνο το συγκεκριμένο τύπο χαρακτήρων. Πχ **bool ctype_digit (string text)** Επιστρέφει true αν όλοι οι χαρακτήρες του ορίσματος είναι αριθμητικοί. **bool ctype_graph (string text)** Επιστρέφει true αν όλοι οι χαρακτήρες του ορίσματος είναι εκτυπώσιμοι, δηλαδή ορατοί (όχι το space).
date	**string date (string format [, int timestamp])** Επιστρέφει μια ημερομηνία με βάση το format
getdate	**array getdate ([int timestamp])** Επιστρέφει έναν πίνακα που περιέχει πληροφορίες για την ημερομηνία του timestamp ή την τρέχουσα αν δεν δοθεί timestamp
time	**int time (void)** Επιστρέφει το τρέχον unix timestamp
localtime	**array localtime ([int timestamp [, bool is_associative]])** Επιστρέφει έναν πίνακα που περιέχει πληροφορίες για την ώρα του timestamp ή την τρέχουσα αν δεν δοθεί timestamp
strtotime	int **strtotime** (string time [, int now]) Μετατρέπει οποιοδήποτε string που περιέχει ημερομηνία ή ώρα με αγγλικό κείμενο σε unix timestamp.
exit	Προκαλεί το τέλος του προγράμματος
die	Τυπώνει ένα μήνυμα και προκαλεί το τέλος του προγράμματος

5.11.2 Συναρτήσεις του MySQL Module

Στον επόμενο πίνακα φαίνονται οι συναρτήσεις της PHP (ένα από τα modules επέκτασης της γλώσσας) για χρήση στην επικοινωνία με MySQL εξυπηρετητή.

Συνάρτηση	**Χρήση**
mysql_affected_rows	Αριθμός γραμμών που επηρεάστηκαν από το προηγούμενο ερώτημα που εκτελέστηκε.
mysql_change_user	Αλλάζει τον χρήση της ενεργής σύνδεσης με τον MySQL server
mysql_client_encoding	Επιστρέφει το όνομα του συνόλου χαρακτήρων της ενεργής σύνδεσης
mysql_close	Κλείνει τη σύνδεση με τον MySQL server
mysql_connect	Ανοίγει μια νέα σύνδεση με τον MySQL server
mysql_create_db	Δημιουργεί μια νέα ΒΔ στον MySQL server
mysql_data_seek	Μετακινεί τον δείκτη σε μια συγκεκριμένη γραμμή στα αποτελέσματα
mysql_db_name	Επιστρέφει το όνομα μίας ΒΔ από ένα σύνολο ΒΔ καθορίζοντας τον αριθμό της γραμμής αυτής που θέλουμε.
mysql_db_query	Στέλνει ένα ερώτημα για εκτέλεση σε μια ΒΔ της ενεργής σύνδεσης
mysql_drop_db	Διαγράφει μια ΒΔ από τον MySQL εξυπηρετητή
mysql_errno	Επιστρέφει τον αριθμό λάθους από την προηγούμενη λειτουργία στον MySQL server
mysql_error	Επιστρέφει το μήνυμα λάθους από την προηγούμενη λειτουργία στον

	MySQL server
mysql_escape_string	Προσθέτει χαρακτήρες διαφυγής στους ειδικούς χαρακτήρες ενός αλφαριθμητικού για χρήση σε ένα ερώτημα
mysql_fetch_array	Φέρνει μία γραμμή από τα αποτελέσματα ενός ερωτήματος σαν έναν πίνακα. Ο πίνακας αυτός είναι και associative (οι δείκτες είναι τα ονόματα των στηλών) και με αριθμητικούς δείκτες (οι δείκτες είναι οι αριθμοί των στηλών)
mysql_fetch_assoc	Φέρνει μία γραμμή από τα αποτελέσματα ενός ερωτήματος σαν έναν associative πίνακα
mysql_fetch_field	Φέρνει πληροφορίες για μια στήλη από τα αποτελέσματα σαν αντικείμενο με συγκεκριμένες ιδιότητες
mysql_fetch_lengths	Φέρνει το μήκος κάθε στήλης της τρέχουσας γραμμής ενός αποτελέσματος
mysql_fetch_object	Φέρνει μία γραμμή από τα αποτελέσματα σαν αντικείμενο με συγκεκριμένες ιδιότητες
mysql_fetch_row	Φέρνει μια γραμμή από τα αποτελέσματα σαν πίνακα με αριθμητικούς δείκτες
mysql_field_flags	Επιστρέφει τις σημαίες που σχετίζονται με ένα συγκεκριμένο πεδίο στο αποτέλεσμα (αν είναι primary_key, auto_increment κλπ)
mysql_field_len	Επιστρέφει το μήκος του συγκεκριμένου πεδίου στο αποτέλεσμα
mysql_field_name	Επιστρέφει το όνομα του συγκεκριμένου πεδίου στο αποτέλεσμα
mysql_field_seek	Θέτει το δείκτη αποτελέσματος στο καθορισμένο πεδίο του αποτελέσματος
mysql_field_table	Επιστρέφει το όνομα του πίνακα στον οποίο είναι το συγκεκριμένο πεδίο
mysql_field_type	Επιστρέφει τον τύπο του συγκεκριμένου πεδίου στο αποτέλεσμα
mysql_free_result	Αποδέσμευσε τη μνήμη από τα δεδομένα του αποτελέσματος
mysql_get_client_info	Επιστρέφει πληροφορίες για τον MySQL πελάτη (έκδοση βιβλιοθήκης)
mysql_get_host_info	Επιστρέφει πληροφορίες για το MySQL host
mysql_get_proto_info	Επιστρέφει πληροφορίες για το MySQL πρωτόκολλο (έκδοση)
mysql_get_server_info	Επιστρέφει πληροφορίες για τον MySQL εξηπηρετητή
mysql_info	Επιστρέφει πληροφορίες για το τελευταίο MySQL ερώτημα
mysql_insert_id	Επιστρέφει το ID (για το πεδίο AUTO_INCREMENT) που δημιουργήθηκε από το προηγούμενο ερώτημα εισαγωγής (Insert query)
mysql_list_dbs	Επιστρέφει τις ΒΔ που βρίσκονται στον MySQL εξυπηρετητή
mysql_list_fields	Επιστρέφει τα πεδία συγκεκριμένου πίνακα μιας ΒΔ που βρίσκεται στον MySQL εξυπηρετητή
mysql_list_processes	Επιστρέφει τις διεργασίες που εκτελούνται στον MySQL εξηπηρετητή
mysql_list_tables	Επιστρέφει τους πίνακες που περιλαμβάνονται σε μια συγκεκριμένη ΒΔ που βρίσκεται στον MySQL εξηπηρετητή
mysql_num_fields	Επιστρέφει τον αριθμό των στηλών του αποτελέσματος ενός ερωτήματος
mysql_num_rows	Επιστρέφει τον αριθμό των γραμμών του αποτελέσματος ενός ερωτήματος
mysql_pconnect	Ανοίγει μια συνεχή (διαρκής) σύνδεση στον MySQL εξυπηρετητή
mysql_ping	Κάνει ping μια σύνδεση ή επανασυνδέεται αν δεν υπάρχει

	επικοινωνία
mysql_query	Στέλνει στον MySQL εξυπηρετητή για εκτέλεση ένα ερώτημα
mysql_real_escape_string	Προσθέτει χαρακτήρες διαφυγής στους ειδικούς χαρακτήρες ενός αλφαριθμητικό ώστε να χρησιμοποιηθεί για ερώτημα. Λαμβάνει υπόψη το τρέχον σύνολο χαρακτήρων (encoding)
mysql_result	Επιστρέφει τα αποτελέσματα ενός ερωτήματος
mysql_select_db	Επιλέγει μια ΒΔ στον MySQL εξυπηρετητή
mysql_stat	Επιστρέφει την τρέχουσα κατάσταση του συστήματος (current system status)
mysql_tablename	Επιστρέφει το όνομα ενός πίνακα μίας ΒΔ από ένα σύνολο πινάκων καθορίζοντας τον αριθμό της γραμμής αυτής που θέλουμε.
mysql_thread_id	Επιστρέφει το ID του thread που σε εξυπηρετεί
mysql_unbuffered_query	Στέλνει ένα ερώτημα χωρίς να περιμένει να έρθει και να μπει στον buffer το αποτέλεσμα.

5.12 Υπερκαθολικές (Superglobal) μεταβλητές

Στην PHP περιλαμβάνονται μια σειρά από Υπερκαθολικές (Superglobal) μεταβλητές οι οποίες είναι άμεσα διαθέσιμες σε οποιοδήποτε σημείο του κώδικά μας και μας παρέχουν διάφορες πληροφορίες για το περιβάλλον εκτέλεσης καθώς και για τα δεδομένα χρήστη κλπ. Αυτές περιλαμβάνουν τις: **$GLOBALS**, **$_SERVER**, **$_POST**, **$_GET**, **$_COOKIE**, **$_FILES**, **$_ENV**, **$_REQUEST**, **$_SESSION**.

Στις επόμενες υποενότητες αναλύονται κάποιες από αυτές. Επίσης υπάρχει η συνάρτηση phpinfo() της PHP που μπορεί να μας επιστρέψει ένα σύνολο από πληροφορίες. Συντάσσεται ως εξής:

int phpinfo ([int what])

και παίρνει σαν όρισμα τις επόμενες σταθερές οι οποίες μάλιστα μπορεί να συνδυαστούν με bitwise OR ώστε να πάρουμε αποτελέσματα πολλών κατηγοριών:

Όνομα Σταθεράς	**Τιμή**	**Περιγραφή**
INFO_GENERAL	**1**	Την γραμμή διαμόρφωσης (κατά την κλήση), την τοποθεσία του php.ini, την ημερομηνία δημιουργίας του περιβάλλοντος, τον Web εξυπηρετητή, το σύστημα κλπ
INFO_CREDITS	**2**	PHP 4 Credits (ομοίως και η **phpcredits()**).
INFO_CONFIGURATION	**4**	Τρέχουσες Local και Master τιμές για τις PHP directives (ομοίως και η **ini_get()**).
INFO_MODULES	**8**	Τα φορτωμένα modules και οι ρυθμίσεις τους (ομοίως και η **get_loaded_modules()**).
INFO_ENVIRONMENT	**16**	Πληροφορίες για τις μεταβλητές περιβάλλοντος (οι οποίες υπάρχουν και στην **$_ENV**
INFO_VARIABLES	**32**	Δείχνει όλες τις προορισμένες μεταβλητές που είδαμε παραπάνω (Environment, GET, POST, Cookie, Server).
INFO_LICENSE	**64**	PHP πληροφορίες άδειας χρήσης
INFO_ALL	**-1**	Δείχνει όλα τα παραπάνω και είναι το προκαθορισμένο όρισμα αν δεν δοθεί.

Παράδειγμα

```
<?php
// Δείχνει όλες τις πληροφορίες όπως και η  phpinfo(INFO_ALL) και η  phpinfo(-1)
phpinfo();

// Δείχνει μόνο τις πληροφορίες των module όπως και η  phpinfo(8)
phpinfo(INFO_MODULES);
?>
```

5.12.1 Αποστολή δεδομένων από τον πελάτη/χρήστη

Τις περισσότερες φορές για να μπορέσουμε να δημιουργήσουμε ένα δυναμικό αποτέλεσμα απαιτούνται δεδομένα εισόδου από τον πελάτη. Τέτοια δεδομένα μπορεί να περιλαμβάνουν στοιχεία προτιμήσεων, στοιχεία εισαγωγής ή ενημέρωσης πινάκων Βάσης Δεδομένων, στοιχεία για την ρύθμιση της επιλογής ή/και τρόπου προβολής των δεδομένων που θα επιστραφούν στον πελάτη κλπ.

Ο συνήθης τρόπος για τη μετάδοση των δεδομένων αυτών από τον πελάτη είναι η χρήση φορμών με τις μεθόδους GET και POST του HTTP καθώς και συνδέσμων που στο URL ακολουθούν ζεύγη πεδίων/τιμών, δηλαδή ουσιαστικά χρήση και πάλι της μεθόδου GET.

Για να μπορούμε να χρησιμοποιήσουμε τα δεδομένα αυτά στα PHP προγράμματα (scripts) πρέπει να μπορούμε να τα λάβουμε μέσα από τον κώδικα της PHP. Ανάλογα με το πώς μας στέλνονται (POST ή GET) τα δεδομένα αυτά, ο τρόπος λήψης στην PHP είναι ο εξής:

- Αν τα δεδομένα στέλνονται με τη μέθοδο GET τότε άμεσα εισάγονται στον πίνακα **$_GET**.
- Αν τα δεδομένα στέλνονται με τη μέθοδο POST τότε άμεσα εισάγονται στον πίνακα **$_POST**.

Συγκεκριμένα πρόκειται για associative πίνακες με δείκτες τα ονόματα των πεδίων και τιμές τις τιμές που συμπλήρωσε ο χρήστης, οι οποίοι είναι Υπερκαθολικές (Superglobal) μεταβλητές, δηλαδή είναι άμεσα διαθέσιμοι παντού στον κώδικά μας. Έτσι αν ο χρήστης πατήσει submit σε μια φόρμα στην οποία υπάρχει ένα πεδίο με το όνομα **name** και στο οποίο συμπλήρωσε, πριν κάνει submit, την τιμή ”**Κατσάλης Νίκος**” τότε στον κώδικά μας μπορούμε να κάνουμε τα εξής:

Αν η φόρμα στάλθηκε με τη μέθοδο GET:

```
$n = $_GET['name'];
echo $n; // τυπώνεται Κατσάλης Νίκος
```

Αν η φόρμα στάλθηκε με τη μέθοδο POST:

```
$k = $_POST['name'];
echo $k; // τυπώνεται Κατσάλης Νίκος
```

Παράδειγμα

Φόρμα χρήστη (πχ αρχείο frm.html)

```
<form action="action.php" method="POST">
 Your name: <input type="text" name="name" />
 Your age: <input type="text" name="age" />
 <input type="submit">
</form>
```

Αρχείο action.php

```
<html>
<body>
Hi <?php echo $_POST["name"]; ?>.
You are
<?php echo $_POST["age"]; ?> years old.
</body>
</html>
```

5.12.2 Πληροφορίες Εξυπηρετητή, Πελάτη και Περιβάλλοντος

Η Υπερκαθολική (Superglobal) μεταβλητή **$_SERVER** είναι ένας associative πίνακας με δείκτες ένα σύνολο strings που αντιπροσωπεύουν στοιχεία του εξυπηρετητή, του πελάτη και του περιβάλλοντος της PHP και τιμές τις τιμές των συγκεκριμένων στοιχείων.
Παράδειγμα:

```
<?php
if (strstr($_SERVER["HTTP_USER_AGENT"], "MSIE")) {
   echo "<center><b>Χρησιμοποιείς Internet Explorer</b></center>";
   echo "Συγκεκριμένα χρησιμοποιείς τον: <br>";
   echo "<b>" . $_SERVER["HTTP_USER_AGENT"]  . "</b>";
} else {
   echo "<h3>Δεν χρησιμοποιείς Internet Explorer</h3>";
   echo "Χρησιμοποιείς τον: <br>";
   echo "<b>" . $_SERVER["HTTP_USER_AGENT"]  . "</b>";
}
?>
```

5.13 Μεθοδολογία Σύνδεσης PHP-MySQL

Για τη σύνδεση με ένα MySQL εξυπηρετητή, την εκτέλεση ερωτημάτων και τον έλεγχο και χρήσ των αποτελεσμάτων υπάρχουν διάφορες προσεγγίσεις που μπορεί κάποιος να χρησιμοποιήσει. Στην υποενότητα αυτή θα παρουσιάσουμε μία προτεινόμενη μεθοδολογία.

Τα βήματα που προτείνονται στην μεθοδολογία αυτή περιλαμβάνουν με τη σειρά:

1. Σύνδεση με τον MySQL εξυπηρετητή
2. Σύνδεση με τη Βάση Δεδομένων του εξυπηρετητή
3. Εκτέλεση Ερωτημάτων
4. Έλεγχος και εκτύπωση αποτελεσμάτων ερωτημάτων
5. Κλείσιμο της σύνδεσης

Ακολουθώντας τα παραπάνω βήματα που αναλύονται στις επόμενες υποενότητες θα μπορείτε πολύ εύκολα να δημιουργήσετε ισχυρές εφαρμογές.

5.13.1 Σύνδεση με τον MySQL εξυπηρετητή

Συνδεόμαστε στον MYSQL server χρησιμοποιώντας τη συνάρτηση **mysql_connect** η οποία παίρνει σαν παραμέτρους το μηχάνημα στο οποίο τρέχει ο MYSQL server, το όνομα χρήστη και τον κωδικό πρόσβασης. Η συνάρτηση αυτή μας επιστρέφει τη σύνδεση που δημιουργείται και την οποία χρειαζόμαστε για τα επόμενα βήματα.

5.13.2 Σύνδεση με τη Βάση Δεδομένων του εξυπηρετητή

Συνδεόμαστε με τη βάση που θέλουμε χρησιμοποιώντας τη συνάρτηση **mysql_select_db** η οποία παίρνει σαν παραμέτρους το όνομα της βάσης και τη σύνδεση που κάναμε στο προηγούμενο βήμα με τον MYSQL server. Η συνάρτηση αυτή μας επιστρέφει μία σύνδεση με τη βάση που δημιουργείται. (Το βήμα αυτό κάνει ότι κάνει και η use database εντολή)

5.13.3 Εκτέλεση Ερωτημάτων

Εκτελούμε ερωτήματα στη βάση χρησιμοποιώντας τη συνάρτηση **mysql_query** η οποία παίρνει σαν παράμετρο το ερώτημα που θέλουμε να εκτελέσουμε. Η συνάρτηση αυτή μας επιστρέφει το αποτέλεσμα του ερωτήματος. Το αποτέλεσμα αυτό αν το ερώτημα είναι επιλογής (SELECT) περιέχει τις γραμμές που επιλέγηκαν.

5.13.4 Έλεγχος και εκτύπωση αποτελεσμάτων ερωτημάτων

Στο προηγούμενο βήμα αν το ερώτημα είναι επιλογής (**SELECT**) μπορούμε χρησιμοποιώντας τη συνάρτηση **mysql_num_rows** που παίρνει σαν παράμετρο το αποτέλεσμα του ερωτήματος να δούμε πόσες γραμμές επιλέχτηκαν. Μετά μπορούμε να κάνουμε μια επανάληψη τόσες φορές όσες μας επιστρέψει η **mysql_affected_rows** και κάθε φορά να καλούμε τη **mysql_fetch_array** η οποία μας επιστρέφει την επόμενη γραμμή (σαν έναν πίνακα με δείκτες τα ονόματα των πεδίων που επιλέχτηκαν ή τον αριθμό της στήλης) την οποία μπορούμε να τυπώνουμε με html.

Αν το ερώτημα είναι εισαγωγής ή ενημέρωσης (**INSERT**, **UPDATE**) μπορούμε χρησιμοποιώντας τη συνάρτηση **mysql_affected_rows** που παίρνει σαν παράμετρο τη σύνδεση στον MYSQL server του βήματος 1 και επιστρέφει πόσες γραμμές επηρεάσθηκαν από το ερώτημα, να δούμε αν πραγματοποιήθηκε και πόσες γραμμές επηρέασε.

5.13.5 Κλείσιμο της σύνδεσης

Κλείνουμε τη σύνδεση που ανοίξαμε στο βήμα 1 χρησιμοποιώντας τη συνάρτηση mysql_close που παίρνει σαν παράμετρο τη σύνδεση αυτή.

5.13.6 Ολοκληρωμένο παράδειγμα των βημάτων

```
/* Βήμα 1: Σύνδεση με τον mysql server που βρίσκεται στο μηχάνημά μας
   (localhost) και η οποία απαιτεί όνομα χρήστη “root” και κωδικό “nikos” */

$mysqluser = "root";
$mysqlpassword = "nikos";

$link = mysql_connect("localhost", $mysqluser, $mysqlpassword);
//αν απέτυχε η σύνδεση βγάλε μήνυμα και τερμάτισε το πρόγραμμα
if (!$link)
   die ('Cannot connect to server for verification : ' . mysql_error());

/* ---------------------------------------------------------------------------------------------------------------- */

/* Βήμα 2: Σύνδεση με τη βάση με όνομα "videoclub" χρησιμοποιώντας την
   προηγούμενη σύνδεση που κάναμε με τον mysql server ($link) */

$mysqldb = "videoclub";
$db_selected = mysql_select_db($mysqldb, $link);
//αν απέτυχε η επιλογή της βάσης (πχ δεν υπάρχει) βγάλε μήνυμα και τερμάτισε
//το πρόγραμμα
if (!$db_selected)
   die ('Cannot connect to database for verification : ' . mysql_error());

/* ---------------------------------------------------------------------------------------------------------------- */
/* Βήμα 3 και 4 για ερώτημα SELECT                                                                                  */

/* Βήμα 3: Εκτέλεση ερωτήματος SELECT στη βάση που συνδεθήκαμε στο προηγούμενο
           βήμα ($db_selected). Υποθέτουμε ότι έχουμε έναν πίνακα με ταινίες και θέλουμε
           να επιλέξουμε όλες τις ταινίες με τίτλο “CRASH”*/

$title = “CRASH”;
$sql = "SELECT fid, title, company FROM films where title='$title'";
$result = mysql_query($sql);
//αν απέτυχε η εκτέλεση του ερωτήματος βγάλε μήνυμα και τερμάτισε
```

```
if (!$result)
   die ('Cannot select from table: ' . mysql_error());

/* Βήμα 4: έλεγχος αποτελεσμάτων ερωτήματος SELECT και εκτύπωσή τους */

$num_results = mysql_num_rows($result);
//αν επιστράφηκαν ταινίες, τύπωσέ τις δημιουργώντας έναν πίνακα σε HTML
if ($num_results > 0)
{
   echo "<table border=1 align=center>";
   echo "<tr><th>fid</th> <th>title</th> <th>company</th> </tr>";
   for ($i=0; $i < $num_results; ++$i)
   {
      /* Επέστρεψε στο $r έναν πίνακα με τα πεδία της επόμενης γραμμής που επιλέχτηκε. */
      $r = mysql_fetch_array($result);
      echo "<tr><td>" . $r['fid'] . "</td><td>" . $r['title'] . "</td><td>" . $r['company'] .
"</td></tr>";
    }
   echo "</table>";
}
else
{
   echo "<h2> Could not find any matching film !!</h2>";
 }

/* ---------------------------------------------------------------------------------------------------------------- */
/* Βήμα 3 και 4 για ερώτημα INSERT                                                                                   */
/* Βήμα 3: Εκτέλεση ερωτήματος στη βάση που συνδεθήκαμε  στο προηγούμενο
              βήμα ($db_selected). Υποθέτουμε ότι έχουμε έναν πίνακα με ταινίες και θέλουμε
              να εισάγουμε μια νέα ταινία με τίτλο "EDISON", της εταιρίας
             "COLUMBIA PICTURES", με περίοδο ενοικίασης 7 ημέρες και κόστος 1.5€ */

$title = "EDISON";
$company = "COLUMBIA PICTURES";
$rentperiod = 7;
$cost = 1.5;
$sql = "insert into films (title, company, rentperiod, cost) values ('$title', '$company',
$rentperiod, $cost)";

$result = mysql_query($sql);
//αν απέτυχε η εισαγωγή του ερωτήματος βγάλε μήνυμα και τερμάτισε
if (!$result)
   die ('Cannot insert in to table : ' . mysql_error());

/* Βήμα 4: έλεγχος αποτελέσματος ερωτήματος INSERT και εκτύπωσή του */
$num = mysql_affected_rows($link);

//αν έγινε η εισαγωγή θα επηρεάστηκε μόνο μια γραμμή (αφού μόνο μία έβαλα)
if ($num != 1)
   echo "<h2> Could not insert film !!</h2>";
else
   echo "<h2> film inserted !!</h2>";

/* ---------------------------------------------------------------------------------------------------------------- */
```

```
/*Βήμα 5: Κλείσιμο σύνδεσης                                              */
mysql_close($link);
```

5.14 Παραδείγματα Εφαρμογών

5.14.1 Παράδειγμα 1

Στο παράδειγμα αυτό υπάρχει μια ιστοσελίδα **example1.html** με μία φόρμα στην οποία δίνεις ένα κείμενο και δύο αριθμούς που καθορίζουν τις διαστάσεις του πίνακα που θέλεις να δημιουργήσεις.

Τα στοιχεία αυτά στέλνονται στο αρχείο **example1.php** με τη μέθοδο POST το οποίο τα διαβάζει χρησιμοποιώντας php και εμφανίζει μια επικεφαλίδα με το κείμενο που έδωσες καθώς και έναν πίνακα με τόσες γραμμές και στήλες όσες οι δύο αριθμοί που έδωσες και περιέχόμενο τον αριθμό της γραμμής και της στήλης.

Αρχείο example1.html

```
<HTML>
<head>
<title>Παράδειγμα 1 HTML Page</title>
<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>
</head>
<BODY>
<p>
<h1>Παράδειγμα 1 PHP Page</h1>
<p>
<FORM action='example1.php' METHOD='post'>
Κείμενο: <INPUT TYPE='text' MAXLENGTH=50 NAME='txt1'>
<p>
Γραμμές: <INPUT TYPE='text' MAXLENGTH=2 NAME='rows'>
<br />
Στήλες: <INPUT TYPE='text' MAXLENGTH=2 NAME='cols'>
<p>
<INPUT TYPE='submit' value='Αποστολή'>
</BODY>
</HTML>
```

Αρχείο example1.php

```
<html>
<head>
<title>Παράδειγμα 1 PHP Page </title>
<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-1'>
</head>
<body>
<?php

/* Παίρνω τις 3 τιμές που μου στάλθηκαν με POST από τον Υπερκαθολικό πίνακα $_POST */
$txt = $_POST['txt1'];
$r = $_POST['rows'];
$c = $_POST['cols'];

/* Τυπώνω το κείμενο επικεφαλίδα 1 */
echo "<h1>" . $txt . "</h1>";
```

```
/* Δημιουργώ έναν πίνακα με $r γραμμές και $c στήλες με κείμενο τον τρέχον αριθμό
   γραμμής και στήλης */
echo "<table border=1>";
for ($i=1;$i<=$r;++$i)
{
		echo "<tr>";
		for ($j=1;$j<=$c;++$j)
		{
				echo "<td>";
				echo "row " . $i . ", column ". $j;
				echo "<td>";
		}
		echo "</tr>";
}
?>
</BODY>
</HTML>
```

5.14.2 Παράδειγμα 2

Στο παράδειγμα αυτό υπάρχει μια ιστοσελίδα **example2.html** με μία φόρμα στην οποία δίνεις το όνομα, το επώνυμο και την ηλικία ενός ατόμου. Τα στοιχεία αυτά στέλνονται στο αρχείο **example2.php**, με τη μέθοδο GET, το οποίο τα διαβάζει χρησιμοποιώντας php και τα εισάγει σε έναν πίνακα επικοινωνώντας με μία βάση MySQL. Ο MySQL server στον οποίο συνδέεται βρίσκεται στον ίδιο Η/Υ (localhost) και ο λογαριασμός του root δεν έχει κωδικό (κάτι που δεν είναι ασφαλές για πραγματική χρήση). Επίσης, για να λειτουργήσει το παράδειγμα, πρέπει στον MySQL server να υπάρχει μία ΒΔ με όνομα **test** στην οποία θα υπάρχει ένας πίνακας με όνομα **testtable** που θα έχει τα εξής πεδία:

onoma char(50), primary key

eponimo char(50)

age int

Αρχείο example2.html

```
<HTML>
<head>
<title>Παράδειγμα 2 HTML Page</title>
</head>
<BODY>
<p>
<h1>Παράδειγμα 2 PHP Page</h1>
<p>
<FORM action='example2.php' METHOD='get'>
Όνομα: <INPUT TYPE='text' MAXLENGTH=50 NAME='firstname'>
<p>
Επώνυμο: <INPUT TYPE='text' MAXLENGTH=50 NAME='lastname'>
<br />
Ηλικία: <INPUT TYPE='text' MAXLENGTH=2 NAME='age'>
<p>
<INPUT TYPE='submit' value='Εισαγωγή'>
</BODY>
</HTML>
```

Αρχείο example2.php

```
<?php
echo "<html>";
echo "<head>";
echo "<title> Παράδειγμα 2 - PHP Page</title>";
echo "</head>";
echo "<body>";

/* Βάζω σε μεταβλητές τα στοιχεία σύνδεσης στον MySQL server (όνομα χρήστη και
κωδικός) καθώς και το όνομα της ΒΔ που θα συνδεθούμε */
        $mysqluser = "root";
        $mysqlpassword = "";
        $mysqldb = "test";


        /* Ελέγχω μήπως κάποιο στοιχείο της φόρμα δεν συμπληρώθηκε ή δεν
          συμπληρώθηκε σωστά. Αν υπάρχει λάθος βάζω ένα μήνυμα στην $error
          την οποία έχω στην αρχή ίση με "", οπότε μετά θα ξέρω αν υπήρχε
          λάθος και ποιο ή ποια ήταν. */
        $error = "";

        /* Ελέγχω μήπως δεν συμπληρώθηκε το όνομα */
        if ($_GET['firstname'] == null)
        {
                $error = $error . "<li> Το όνομα είναι κενό. </li>";
        }
        else
                $on = $_GET['firstname'];

        /* Ελέγχω μήπως δεν συμπληρώθηκε το επώνυμο */
        if ($_GET['lastname'] == null)
        {
                $error = $error . "<li> Το επώνυμο είναι κενό. </li>";
        }
        else
                $ep = $_GET['lastname'];


        /* Ελέγχω μήπως δεν συμπληρώθηκε η ηλικία ή μήπως δεν είναι αριθμός > 0*/
        if ($_GET['age'] == null)
        {
                $error = $error . "<li> Η ηλικία είναι κενή. </li>";
        }
        else
        {
                $ilikia = $_GET['age'];

                if (ctype_digit($ilikia) == 0)
                {
                    $error = $error . "<li> Η ηλικία δεν είναι αποδεκτός αριθμός. </li>";
                }
                else
                {
```

```
                        if ($ilikia <= 0)
                          $error = $error . "<li> Η ηλικία πρέπει να είναι μεγαλύτερη από
0</li>";
                }
        }

        /* Αν συνέβει λάθος τότε η μεταβλητή $error δεν είναι κενή */
        if (trim($error)!="")
        {
                echo "<ol>";
                echo $error;
                echo "</ol>";
                echo "<br><b> Η εισαγωγή απέτυχε !! </b>";
        }
        else  /* αλλιώς ξεκινάω τη διαδικασία για εισαγωγή στην Β.Δ. */
        {
                /* Συνδέομαι με τον mysql server */
                $link = mysql_connect("localhost", $mysqluser, $mysqlpassword);
                if (!$link)
                        die ('Cannot connect to server for verification : ' . mysql_error());

                /* Επιλέγω τη Βάση Δεδομένων από τον mysql server */
                $db_selected = mysql_select_db($mysqldb, $link);
                if (!$db_selected)
                  die ('Cannot connect to database for verification : ' . mysql_error());

            /* Επιλέγω όλα τα στοιχεία από τον testtable με το όνομα που μου δόθηκε */
                $sql = "SELECT * FROM testtable where onoma='$on'";
                $result = mysql_query($sql);

                if (!$result)
                        die ('Cannot select from table for verification : ' . mysql_error());

                /* Παίρνω τον αριθμό των εγγραφών που επιλέχτηκαν */
                $num = mysql_num_rows($result);

                /* Αν επιλέχτηκε έστω και 1 πάει να πει ότι υπάρχει και δεν πρέπει να
                  το ξαναεισάγω αφού το όνομα είναι κλειδί */
                if ($num >= 1)
                {
                        /* close connection */
                        mysql_close($link);
                        $error = $error . "<li> Υπάρχει ήδη αυτό το όνομα. </li>";
                        echo "<ol>";
                        echo $error;
                        echo "</ol>";
                        echo "<br><b> Η εισαγωγή απέτυχε. </b>";
                }
                else /* Αλλιώς (δηλαδή δεν υπάρχει ήδη το όνομα) το εισάγω */
                {
                        $sql = "insert into testtable (onoma, eponimo, age) values ('$on',
'$ep', $ilikia)";
                        $result = mysql_query($sql);
                        if (!$result)
                                die ('Cannot insert in to table : ' . mysql_error());
```

```
/* Κοιτάζω πόσες εγγραφές εισήχθηκαν */
$num = mysql_affected_rows($link);

if ($num != 1)
{
        die ('Cannot insert product : ' . mysql_error());
}
else
{
        /* close connection */
        mysql_close($link);
        echo "<b> Η εισαγωγή πέτυχε !!</b> για την εγγραφή με
τα στοιχεία: <br>";
        echo $on . "<br>";
        echo $ep . "<br>";
        echo $ilikia . "<br>";

}
                }
        }
echo "</BODY>";
echo "</HTML>";
?>
```

5.14.3 Παράδειγμα 3

Στο παράδειγμα αυτό υπάρχει μια ιστοσελίδα **example3.html** με μία φόρμα στην οποία δίνεις το όνομα ή μια έκφραση με την οποία συγκρίνεται το όνομα του ατόμου (πχ % για όλους ή N% για όσους το όνομα αρχίζει από N). Τα στοιχεία αυτά στέλνονται στο αρχείο **example3.php**, με τη μέθοδο POST, το οποίο τα διαβάζει χρησιμοποιώντας php και κάνει ένα ερώτημα στον MySQL server και επιστρέφει (προβάλλει) όλες τις εγγραφές που ταιριάζουν δημιουργώντας έναν πίνακα σε HTML όπου σε κάθε γραμμή είναι τα στοιχεία μίας από τις εγγραφές που ταιριάζουν στην έκφραση αναζήτησης.

O Mysql server στον οποίο συνδέεται βρίσκεται στον ίδιο H/Y (localhost) και ο λογαριασμός του root δεν έχει κωδικό (κάτι που δεν είναι ασφαλές για πραγματική χρήση). Επίσης, για να λειτουργήσει το παράδειγμα, πρέπει στον MySQL server να υπάρχει μία ΒΔ με όνομα test στην οποία θα υπάρχει ένας πίνακας με όνομα testtable που θα έχει τα εξής πεδία:

onoma char(50), primary key

eponimo char(50)

age int

Αρχείο example3.html

```
<HTML>
<head>
<title>Παράδειγμα 3 HTML Page</title>
</head>
<BODY>
<p>
<h1>Παράδειγμα 3 PHP Page</h1>
<p>
```

```
<FORM action='example3.php' METHOD='post'>
Όνομα ή λογική έκφραση που θα συγκριθεί με το πεδίο όνομα: <INPUT TYPE='text'
MAXLENGTH=50 NAME='firstname'>
<p>
<INPUT TYPE='submit' value='Αναζήτηση-Προβολή'>
</BODY>
</HTML>
```

Αρχείο example3.php

```
<?php

echo "<html>";
echo "<head>";
echo "<title>Παράδειγμα 3 - PHP Page</title>";
echo "</head>";
echo "<body>";


/* Βάζω σε μεταβλητές το όνομα χρήστη, τον κωδικό και το όνομα της ΒΔ στην
  οποία θα συνδεθώ */
          $mysqluser = "root";
          $mysqlpassword = "";
          $mysqldb = "test";


          /* Ελέγχω μήπως κάποιο στοιχείο της φόρμα δεν συμπληρώθηκε ή δεν
            συμπληρώθηκε σωστά. Αν υπάρχει λάθος βάζω ένα μήνυμα στην $error
            την οποία έχω στην αρχή ίση με "", οπότε μετά θα ξέρω αν υπήρχε
            λάθος και ποιο ή ποια ήταν. */
          $error = "";

          /* Ελέγχω μήπως δεν συμπληρώθηκε το όνομα*/
          if ($_POST['firstname'] == null)
          {
                    $error = $error . "<li>Το όνομα είναι κενό.</li>";
          }
          else
                    $on = $_POST['firstname'];

          /* Αν συνέβη λάθος τότε η μεταβλητή $error δεν είναι κενή */
          if (trim($error)!="")
          {
                    echo "<ol>";
                    echo $error;
                    echo "</ol>";
                    echo "<br><b>Η αναζήτηση-προβολή απέτυχε !!</b>";
          }
          else  /* αλλιώς ξεκινάω τη διαδικασία για αναζήτηση-προβολή */
          {
                    /* Συνδέομαι με τον mysql server */
                    $link = mysql_connect("localhost", $mysqluser, $mysqlpassword);
                    if (!$link)
                              die ('Cannot connect to server for verification : ' . mysql_error());
```

```
/* Επιλέγω τη Βάση Δεδομένων από τον mysql server */
$db_selected = mysql_select_db($mysqldb, $link);
if (!$db_selected)
die ('Cannot connect to database for verification : ' . mysql_error());

/* Επιλέγω όλα τα στοιχεία από τον testtable με το όνομα που μου
  δόθηκε, χρησιμοποιώντας like */
$sql = "SELECT * FROM testtable where onoma like '$on'";
$result = mysql_query($sql);

if (!$result)
        die ('Cannot select from table for verification : ' . mysql_error());

/* Παίρνω τον αριθμό των εγγραφών που επιλέχτηκαν */
$num = mysql_num_rows($result);

/* Αν επιλέχτηκε έστω και 1 δημιουργώ έναν πίνακα HTML και
  την/τις προβάλω */
if ($num >= 1)
{
        echo "<center>Ταιριάζουν " . $num . " εγγραφές:</center>";
        echo "<table>";
        echo "<tr><th>Όνομα</th> <th>Επώνυμο</th>
<th>Ηλικία</th></tr>";

        for ($i=0; $i < $num; ++$i)
        {
                $r = mysql_fetch_array($result);
                echo "<tr><td>" . $r['onoma'] . "</td> <td>" .
$r['eponimo'] . "</td> <td>" . $r['age'] . "</td></tr>";
        }
        echo "</table>";

        /* close connection */
        mysql_close($link);
}
else /* Αλλιώς (δηλαδή δεν ταιριάζει καμία) */
{

        /* close connection */
        mysql_close($link);
        echo "<b>Δεν υπάρχει εγγραφή που να ταιριάζει με το
όνομα/λογική έκφραση που δώσατε !<br>";

}
    }
echo "</BODY>";
echo "</HTML>";
?>
```

6 Αναφορές

- *PHP manual from PHP Documentation Group (http://www.php.net/docs.php)*
- *MySQL Documentation (http://dev.mysql.com/doc/)*
- *Using PHP with MySQL, articles and documentation (http://dev.mysql.com/usingmysql/php/)*